H. & M. Tornsdorf

TIPS & TRUCS

DOS 5

Grensverleggende gebruikersgeheimen

DATA BECKER
NEDERLANDS

Harry Parren 19

TIPS & TRUCS

DOS 5

Grensverleggende gebruikersgeheimen

H & M. Tornsdorf

TIPS & TRUCS

DOS 5

Grensverleggende gebruikersgeheimen

DATA BECKER
NEDERLANDS

Oorspronkelijke titel
DOS 5.0 Tips & Tricks
© 1991 Data Becker GmbH, Düsseldorf
Voor de Nederlandse vertaling
© 1991 A.W. Bruna Uitgevers B.V., Utrecht
Vertaling
LINE UP tekstprodukties

Omslagontwerp
Studio Herman Bade

ISBN 90 229 3778 X
NUGI 852

Alle rechten voorbehouden. Niets uit deze uitgave mag worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door middel van fotokopieën, opnamen, of op welke wijze ook, zonder voorafgaande schriftelijke toestemming van de uitgever.

Voor zover het maken van kopieën uit deze uitgave is toegestaan op grond van artikel 168 Auteurswet j° het Besluit van 20 juni 1974, Stb. 351, zoals gewijzigd bij het Besluit van 23 augustus 1985, Stb. 471 en artikel 17 Auteurswet 1912, dient men de daarvoor wettelijk verschuldigde vergoedingen te voldoen aan de Stichting Reprorecht (Postbus 882, 1180 AW Amstelveen). Voor het overnemen van gedeelte(n) uit deze uitgave in bloemlezingen, readers en andere compilatiewerken (artikel 16 Auteurswet 1912) dient men zich tot de uitgever te wenden.

Ondanks alle aan de samenstelling van de tekst bestede zorg, kan noch de redactie noch de uitgever aansprakelijkheid aanvaarden voor eventuele schade, die zou kunnen voortvloeien uit enige fout, die in deze uitgave zou kunnen voortkomen.

In dit boek komen namen voor van gedeponeerde handelsmerken. Deze namen zijn in de tekst niet voorzien van een handelsmerk-symbool, omdat ze slechts fungeren als aanduiding van de besproken produkten. Hierbij wordt op geen enkele wijze getracht inbreuk te maken op de rechten van de handelsmerkhouder.

Wilt u op de hoogte worden gehouden van onze nieuwe uitgaven, stuur dan een briefkaart naar A.W. Bruna Uitgevers B.V., Postbus 8411, 3503 RK Utrecht. Vergeet niet uw naam en adres te vermelden.

Inhoud

Inleiding

- Een snelle overstap op MS-DOS 5.0; direct alle nieuwe mogelijkheden gebruiken.
- Met behulp van het geheugenbeheer van MS-DOS 5.0 het maximum uit uw werkgeheugen halen.
- Een op alfabetische volgorde gesorteerde lijst met **alle** bestanden op een harde schijf, onafhankelijk van de directory-structuur.
- Uitgebreide zoekfuncties voor bestanden, op naam maar ook op trefwoord.
- De mogelijkheid alle bestanden met de extensie .BAK of .SIK in één keer te wissen in alle directories van de harde schijf.
- Een editor voor de opdrachtregel, waarmee u zelfs interne DOS-opdrachten kunt wijzigen.
- Beveiligingsroutines voor het systeem of bepaalde bestanden.
- Wisroutines die het herstel van per ongeluk gewiste gegevens mogelijk maken.

De vereiste voorkennis

In dit boek richten we ons tot lezers die de eerste beginselen van DOS onder de knie hebben. Het is niet geschreven voor professionele gebruikers van DOS, maar voor hen die nog wel wat hulp kunnen gebruiken bij het maken van bijvoorbeeld batchbestanden. Daarnaast geven we een aantal tips voor het gebruik van QBASIC en DEBUG bij het oplossen van problemen.

Ook als u zich niet voor alle details van de tips en aanwijzingen interesseert, biedt dit boek u tal van mogelijkheden om het dagelijkse werk met MS-DOS te verlichten.

Tip In dit boek gaan we uit van MS-DOS van Microsoft. IBM, maar ook een paar andere producenten van hardware, maken eigen versies van MS-DOS in licentie, die in details kunnen afwijken. Waar nodig zullen we daar aandacht aan besteden.

Hoe u het beste met dit boek kunt werken

Leest u eerst hoofdstuk 1 in zijn geheel door. Daar vindt u alles beschreven wat u moet weten om de aanwijzingen in het vervolg te kunnen begrijpen en gebruiken. Uit de daarna volgende hoofdstukken kunt u naar believen die onderwerpen uitzoeken, die u het meest interesseren. De tips en de trucs aan het eind van het boek staan los van elkaar; u kunt ze afzonderlijk toepassen. Raadpleeg het register en het overzicht van tips en trucs om oplossingen te vinden voor speciale problemen.

1 De vereiste voorkennis

We zullen proberen dit hoofdstuk zo kort mogelijk te houden, omdat dit boek niet is bedoeld als inleiding tot MS-DOS. Toch is dit hoofdstuk onvermijdelijk, omdat we de onderwerpen onafhankelijk van elkaar willen behandelen. We stellen ons ten doel de hoofdstukken zo te schrijven, dat u na lezing van dit hoofdstuk zonder problemen een willekeurig onderwerp kunt volgen.

Voor alle aanwijzingen en achtergronden die in een bepaald hoofdstuk niet behandeld kunnen worden, geven we uiteraard verwijzingen. Voor het overige gaan we ervan uit dat u beschikt over de kennis van een MS-DOS boek voor beginners.

1.1 Inleiding tot het werken met batchbestanden

We maken bij het uitvoeren van de tips en trucs veel gebruik van zogenaamde batchbestanden. Ze zijn makkelijk aan te maken en vereenvoudigen het werken met MS-DOS aanzienlijk. Omdat ze een grote rol spelen in dit boek, beginnen we met een overzicht van de hoofdzaken van het werken met batchbestanden.

We maken daarbij onderscheid tussen gebruikers die nog nooit met batchbestanden hebben gewerkt en gebruikers die over een basiskennis beschikken. Voor de beginners volgt hieronder een vrij uitvoerige, maar eenvoudig gehouden inleiding. Daarin behandelen we het wezen en de betekenis van batchbestanden en leggen we de belangrijkste regels uit. Voor een beter begrip hebben we die paragraaf geschreven alsof het een deel van een boek voor beginners was.

De daaropvolgende paragraaf behandelt wat meer ingewikkelde regels, die nodig zijn voor een optimaal gebruik van batchbestanden. Na lezing van die paragraaf bent u zover dat u zonder problemen aan de slag kunt met de voorbeeldprogramma's van dit boek. Voor wie batchbestanden geen geheimen meer hebben, hebben we een overzichtje gemaakt van de kennis die noodzakelijk is. Aan de hand daarvan kunt u dan snel zien, wat u bekend of onbekend voorkomt.

1.2 Batchbestanden voor beginners

Hebt u nog nooit met batchbestanden gewerkt, dan raden we u aan deze paragraaf eerst door te nemen.

Het belang van batchbestanden

Het besturingssysteem MS-DOS biedt weliswaar een schat aan bewerkingsmogelijkheden, maar die worden niet erg gebruikersvriendelijk aangeboden. Zo zult u na het invoeren van een opdracht via de opdrachtregel eerst moeten wachten tot de prompt weer beschikbaar is, voordat u de volgende opdracht kunt invoeren. Batchbestanden lossen dit probleem voor u op. Het Engelse woord batch betekent vrij vertaald zoveel als 'stapel' en een batchbestand is dan ook niets meer dan een stapeltje MS-DOS-opdrachten. Batchbestanden zijn altijd herkenbaar aan de extensie .BAT. U activeert een batchbestand, net als een bestand met de extensie .COM of .EXE, door alleen de bestandsnaam in te voeren. De extensie laat u dus achterwege.

Een eerste batchbestand maken

We zullen het principe van het batchbestand proberen uit te leggen aan de hand van een voorbeeld. Een van de opdrachten van MS-DOS 5.0 is de opdracht DOSKEY, waarachter een programma schuilgaat dat onder andere eerder gegeven opdrachten kan herhalen. Start het programma door achter de prompt te typen:

```
DOSKEY
```

Sluit de invoer af met Enter. In het vervolg zullen we niet steeds aangeven dat u de invoer moet afsluiten met Enter. Ongetwijfeld hebt u al eens na het invoeren van een opdracht verwachtingsvol naar het scherm zitten staren, waarop niets te zien was dan de opdracht zelf en een knipperende cursor... tot het moment dat u doorkreeg dat u opdrachten pas activeert door op Enter te drukken.

Tip Als u de opdracht DOSKEY al eerder hebt ingevoerd nadat u het systeem startte, kunt u de opdrachtbuffer wissen door op ALT-F7 te drukken.

In dit voorbeeld is de hoofddirectory actief en wilt u een subdirectory bekijken. Daarna keert u terug naar de hoofddirectory.

De eerste stap is het activeren van de subdirectory, in dit geval de directory TEMP. Voer de volgende opdracht in:

```
CD TEMP
```

Als uw computer de foutmelding 'Invalid directory' (ongeldige directory) op het scherm plaatst, bestaat de directory nog niet. In dat geval maakt u de directory TEMP aan met de opdracht:

```
MD TEMP
```

Daarna drukt u op Alt-F7 om de opdrachtbuffer leeg te maken. Nu beschikt u over de directory en kunt u overnieuw beginnen met de opdracht 'CD TEMP'.

Plaats vervolgens een overzicht van de inhoud van de subdirectory op uw scherm met de opdracht:

```
DIR
```

Daarna keert u terug naar de hoofddirectory met:

```
CD \
```

Dit zijn de drie opdrachten die u nodig hebt voor het uitvoeren van het voorbeeld.

Drukt u op F7, dan verschijnt een overzichtje van de drie opdrachten op het scherm. Samen vormen zij al bijna een batchbestand. Voordat het echter zover is, moeten de opdrachten nog worden opgeslagen in een bestand. Dat kan ook met behulp van het programma DOSKEY, dat u al in het geheugen hebt geladen. Voer daartoe de volgende opdracht in:

```
DOSKEY /HISTORY > TEST.BAT
```

Met behulp van deze opdracht maakt u het bestand TEST.BAT aan waarin alle tot nu toe ingevoerde opdrachten worden opgeslagen. Ook deze laatste dus, waarmee u het bestand aanmaakt. Om die regel weer te verwijderen roept u de *editor* EDIT op samen met de naam van het nieuwe bestand:

```
EDIT TEST.BAT
```

De editor wordt gestart en het bestand wordt meteen geladen, zodat u dadelijk aan de slag kunt. Met behulp van de toets cursor omlaag plaatst u de cursor op de onderste regel. U markeert de regel door de Shift-toets ingedrukt te houden en op cursor naar rechts te drukken. Herhaal die handeling tot de hele regel is gemarkeerd. Drukt u nu op Del, dan wordt de regel gewist. U moet het bestand opnieuw opslaan. Kies met Alt-F en S het menu File en Save. Verlaat de editor met Alt-F en X (File en Exit).

Het batchbestand is nu klaar. U kunt het uitproberen door de volgende opdracht in te voeren:

```
TEST
```

Zoals het een goede computer betaamt, neemt hij u nu het werk uit handen en voert hij de drie opdrachten die u eerst stuk voor stuk met de hand moest invoeren, keurig netjes uit. U hebt uw eerste batchbestand gemaakt.

Het bestand AUTOEXEC.BAT

Waarschijnlijk hebt u, ook al wist u dat niet, al talloze malen batchbestanden gebruikt. Het bestand AUTOEXEC.BAT is namelijk ook een batchbestand. Het wordt geactiveerd zodra u uw computer start en maakt het systeem klaar voor gebruik door een aantal DOS-opdrachten op te roepen. Wat er allemaal gebeurt, hangt af van de opdrachten in het bestand. Bekijkt u het bestand eens door de volgende opdracht in te voeren:

```
TYPE AUTOEXEC.BAT
```

De inhoud van het bestand wordt nu op uw scherm geplaatst. Te midden van alle opdrachten herkent u misschien de volgende, die al of niet in gewijzigde vorm, veel worden gebruikt:

```
PATH C:\DOS
PROMPT $P$G
KEYB NL
```

De andere opdrachten zijn op dit moment niet van belang. De drie bovenstaande kunt u zelf ook invoeren om het werken met de computer makkelijker te maken.

De opdracht PATH zorgt voor een zoekpad naar de directory DOS, waarin alle DOS-opdrachten staan. Het besturingssysteem kan met behulp van het zoekpad altijd de DOS-opdracht vinden die u invoert, ook al is een andere directory op dat moment actief. De opdracht PROMPT zorgt voor de weergave van de prompt waaraan u kunt zien dat het systeem klaar is voor invoer. De gegevens die worden voorafgegaan door een dollarteken bepalen hoe de prompt wordt weergegeven. De opdracht NL past het toetsenbord aan voor het gebruik van Nederlandse tekens.

U zou alle opdrachten ook met de hand kunnen invoeren, steeds als u het systeem start, maar dat zou u snel vervelen en het druist in tegen het wezen van de computer. Erg handig dus, dat er bestanden mogelijk zijn als het bestand AUTOEXEC.BAT.

Dat het bestand een batchbestand is met opdrachten die u steeds opnieuw kunt uitvoeren, kunt u eenvoudig controleren. Typ het volgende:

```
AUTOEXEC
```

Wacht op het resultaat. Afhankelijk van de opdrachten in het bestand kunnen er foutmeldingen op uw scherm verschijnen, omdat het programma opdrachten probeert uit te voeren, die al zijn uitgevoerd.

Batchbestanden invoeren en wijzigen

Het principe van batchbestanden is u hopelijk duidelijk geworden in de vorige paragraaf. We zullen ons nu bezighouden met het aanmaken en wijzigen van batchbestanden.

U kunt batchbestanden, net zoals veel andere bestanden, op meer dan één manier aanmaken. De hoofdzaak daarbij is dat het bestand wordt opgeslagen in ASCII-code. Grofweg kunt u het bestand op drie manieren aanmaken:

- met de DOS-opdracht COPY CON
- met de DOS-editor EDIT
- met een tekstverwerker die bestanden kan opslaan in ASCII-code.

We zullen de drie methoden hieronder kort toelichten.

COPY CON

1 Roep de opdracht COPY CON op samen met de naam die het bestand moet krijgen en de extensie .BAT. Voor het aanmaken van een batchbestand met de naam WP.BAT, voert u dus in:

```
COPY CON WP.BAT
```

2 U voert de DOS-opdrachten van het batchbestand per regel in en sluit iedere regel af met Enter. Om de tekstverwerker WP op te roepen, voert u dus in:

```
C:\WP\WP
```

Maakt u een typefout, dan kunt u met de Backspace-toets terug langs de actuele regel naar het foutieve teken. Alle tekens die daarna kwamen, worden gewist; die

zult u opnieuw moeten invoeren. Hebt u eenmaal een regel afgesloten met Enter en bevindt de cursor zich dus op een nieuwe regel, dan kunt u geen wijzigingen meer aanbrengen in de voorgaande regel.

3 Nadat u de laatste regel hebt ingevoerd en op Enter hebt gedrukt, slaat u het bestand op met de toetsencombinatie Ctrl-Z of de functietoets F6. Het bestand wordt opgeslagen in de actuele directory.

4 Kopieer het bestand naar een directory die is opgenomen in de opdracht PATH, zodat u het bestand vanuit alle directories kunt activeren.

Het uiteindelijke resultaat is dat u de tekstverwerker WordPerfect vanuit een willekeurige directory kunt starten. U hoeft niet eerst de directory met de programmabestanden van WordPerfect te activeren.

COPY CON leent zich goed voor het aanmaken van kleine batchbestanden met opdrachten waarvan u de syntaxis goed kent. Worden de opdrachten wat ingewikkelder, dan maakt u al gauw foutjes, waarna u het hele batchbestand opnieuw moet aanmaken met COPY CON of gebruik moet maken van de hieronder staande methoden om een batchbestand te bewerken met de editor van MS-DOS.

EDIT

De editor van MS-DOS 5.0 is een moderne tekstverwerker die gebruik maakt van venstertechnieken en daardoor zeer gebruikersvriendelijk is.

1 Roep de editor op met de opdracht:

```
EDIT NAAM.BAT
```

Achter de opdracht waarmee u de editor oproept, voert u de naam en de extensie in van het bestand dat u aan wilt maken.

2 Nadat het programma is gestart, wordt het grootste deel van het scherm ingenomen door het werkblad van de editor. U kunt direct beginnen te typen op de plaats van de cursor. Omdat de opdrachten van een batchbestand stuk voor stuk op een afzonderlijke regel moeten staan, sluit u iedere opdracht af met Enter.

3 Na het invoeren slaat u het bestand als volgt op. Met de toetsencombinatie Alt-F opent u het menu File. Druk vervolgens op S (Save). Bij het opslaan krijgt het bestand de naam mee die u hebt opgegeven.

4 Voordat u het batchbestand kunt testen, moet u de editor nog verlaten. U opent opnieuw het menu File met Alt-F en drukt op X voor Exit. De editor wordt gesloten en de vertrouwde DOS-prompt verschijnt weer op uw scherm.

Tekstverwerkers

Bent u in het bezit van een tekstverwerker waarmee u goed uit de voeten kunt, dan kunt u die gebruiken om een batchbestand aan te maken. De tekstverwerker moet dan wel een functie hebben om bestanden op te slaan in ASCII-formaat.

De bekende professionele tekstverwerkers beschikken allemaal over zo'n functie. Of dat ook opgaat voor uw tekstverwerker, zult u zelf moeten uitzoeken. Sla de tekst op met de opdracht die daarvoor volgens u in aanmerking komt; verlaat de tekstverwerker en breng de inhoud van het batchbestand op uw scherm met de opdracht TYPE. Ziet de tekst er nu net zo uit als wanneer u hem maakt met COPY CON of EDIT, dan is alles in orde en kunt u dus op die manier van uw vertrouwde tekstverwerker gebruik maken. Verschijnen er allemaal rare tekens in de tekst, dan duiden die op allerlei codes, die het bestand ongeschikt maken als batchbestand.

In principe kunt u nu batchbestanden aanmaken. We gaan verder met een aantal voorbeelden van batchbestanden die u het werken met DOS al wat makkelijker zullen maken.

Voorbeeld: Een batchbestand om opdrachten in te voeren

De batchbestanden dienen als basis voor veel tips en trucs. Het onderstaande minuscule batchbestandje is in feite de eerste daarvan. Nogal wat DOS-opdrachten vereisen het nodige typewerk. Zet u zo'n opdracht in een batchbestand, dan hoeft u zich geen zorgen meer te maken om de juiste syntaxis. Het is wel zaak zo'n batchbestand een naam te geven die iets zegt over de inhoud of de functie ervan. Mocht deze manier van werken bij u aanslaan, en zou u ze eenvoudigweg nummeren (1.BAT, 2.BAT, enzovoort), dan bent u bij 14.BAT waarschijnlijk al lang vergeten waarvoor 6.BAT ook weer was bedoeld.

In het voorbeeld gaan we uit van de opdracht DIR /O:D. De opdracht verzorgt een overzicht van de inhoud van de actuele directory, gesorteerd op datum. Met behulp van COPY CON maakt u in een ommezien het volgende batchbestand:

```
COPY CON DS.BAT
```

DS is als naam gekozen, omdat de letters staan voor: 'Datum, Sorteren'. Op de volgende regel typt u de tekst van de opdracht:

```
DIR /O:D
```

U brengt de cursor met Enter naar de volgende regel en slaat het bestand op met Ctrl-Z of F6.

Kopieer het bestand naar een voor de hand liggende directory en maak een zoekpad aan naar die directory. Aan het eind van deze paragraaf komt de opdracht PATH nog wat uitvoeriger aan de orde. U kunt nu de opdracht gebruiken, ongeacht de actuele directory. U typt achter de DOS-prompt 'DS' en sluit af met Enter. Het resultaat is een op datum gesorteerd overzicht van de inhoud van de actuele directory.

Voorbeeld: een batchbestand om een directory te activeren

Bij dit tweede voorbeeld gaan we ervan uit dat u op uw harde schijf een hiërarchie hebt aangebracht in directories en subdirectories. Veronderstel dat u al werkt met vier niveaus:

```
c:\teksten\prive\sport\schaken
```

Het is vervelend om steeds het hele pad te moeten typen. Het batchbestandje SCHAKEN.BAT neemt het werk van u over. Maak het bestand weer aan met de opdrachten:

```
COPY CON SCHAKEN.BAT
CD C:\teksten\prive\sport\schaken
```

Sla het bestand weer op met Ctrl-Z of F6. MS-DOS maakt bij dit soort opdrachten geen onderscheid tussen hoofdletters en kleine letters; u kunt beide door elkaar gebruiken.

Wilt u van de gelegenheid gebruik maken om ook direct een overzicht te krijgen van alle bestanden met de extensie .TXT, dan voegt u de volgende regel toe voordat u het bestand opslaat:

```
DIR *.TXT
```

Het bestand is in de eigenlijke zin van het woord nu pas een batchbestand, omdat er twee opdrachten op elkaar zijn gestapeld.

Voorbeeld: een batchbestand om programma's te starten

Hiervoor hebben we al een keer een voorbeeld gegeven van een batchbestand waarmee u de tekstverwerker WordPerfect kunt oproepen. Na de opdracht:

```
COPY CON W.BAT
```

typt u bijvoorbeeld:

```
C:\PROGS\WP\WP
```

U slaat het bestand weer op de gebruikelijke manier op. Uiteraard geeft u vóór de programmanaam het zoekpad dat voor u van toepassing is. U kunt nu ook meer batchbestanden aanmaken voor het starten van WordPerfect met andere startopties. De onderstaande opdrachten maken een batchbestand aan dat WordPerfect start en direct het bestand TEKST.TXT laadt:

```
COPY CON WTEKST.BAT
C:\PROGS\WP\WP TEKST.TXT
```

In plaats van WTEKST.BAT kunt u uiteraard een andere naam kiezen die u beter kunt onthouden of makkelijker kunt typen.

Voorbeeld: programma's koppelen in een batchbestand

Bij sommige programma's, bijvoorbeeld uw tekstverwerker WORD, maakt u gebruik van de muis en wilt u de muis-driver oproepen voordat u het programma start. Maak daarvoor een batchbestand aan, bijvoorbeeld onder de naam WMUIS.BAT. Ervan uitgaand dat de muis-driver is opgeslagen in de directory MUIS, luiden de opdrachten van het batchbestand:

```
C:\MUIS\MUIS
C:\PROGS\WORD\WORD
```

Hebt u WORD al een keer gestart met WMUIS.BAT en wilt u het programma een tweede keer starten, dan hoeft u de muis-driver niet meer te installeren. U kunt WORD dan oproepen met een ander batchbestand. U kiest dus het batchbestand dat bij de situatie past.

Voorbeeld: een backup maken met een batchbestand

We zullen het aantal opdrachten in het batchbestand nog verder uitbreiden. We maken een batchbestand aan voor het maken van een backup van een subdirectory: alle bestanden in die directory worden voor de zekerheid gekopieerd naar een diskette. De volgende handelingen worden door het batchbestand verricht:

1 Een directory wordt samen met alle subdirectories van die directory opgeslagen. De parameter /L zorgt voor een protocolbestand met de namen van alle gekopieerde bestanden.

2 Het protocolbestand wordt naar de backup-diskette gekopieerd onder een nieuwe naam.

3 Het protocolbestand wordt naar de betrokken directory op de harde schijf gekopieerd, weer onder een nieuwe naam, die betrekking heeft op de naam van de directory.

4 Het protocolbestand in de hoofddirectory wordt gewist.

De code 'B50' die in de opdrachten wordt gebruikt, duidt erop dat de backup is gemaakt met behulp van MS-DOS 5.0. 'SUB' staat voor de naam van de betrokken subdirectory en zal dus steeds moeten worden aangepast als u een backup maakt. 'DISKLOG' is de naam van een subdirectory op de harde schijf, die uitsluitend protocolbestanden bevat.

Wilt u het batchbestand uitproberen, maak dan eerst de beide directories SUB en DISKLOG aan. De inhoud van het batchbestand, B50SUB.BAT, is als volgt:

```
BACKUP C:\SUB\*.* A: /S/L
COPY C:\BACKUP.LOG A:\B50SUB.TXT
COPY C:\BACKUP.LOG C:\DISKLOG\B50SUB.TXT
DEL C:\BACKUP.LOG
```

Dit batchbestand begint er al aardig op te lijken. Het heeft één nadeel: u kunt het in zijn huidige vorm alleen gebruiken voor de directory SUB. Wilt u van een andere directory een vergelijkbare backup maken, dan zult u eerst met behulp van EDIT of een tekstverwerker de naam SUB moeten vervangen door de naam van die directory.

Gelukkig is er ook een manier om het batchbestand zodanig te schrijven dat u steeds de gelegenheid krijgt een naam in te voeren, nadat u het batchbestand start.

Voorbeeld: een batchbestand met een variabele

U hoeft weinig te doen om het bovenstaande batchbestand zodanig te wijzigen dat hij niet alleen is gekoppeld aan de naam SUB:

```
BACKUP C:\%1\*.* A: /S/L
COPY C:\BACKUP.LOG A:\B50%1.TXT
COPY C:\BACKUP.LOG C:\DISKLOG\B50%1.TXT
DEL C:\BACKUP.LOG
```

Sla het bestand bijvoorbeeld op onder de naam B50.BAT. U kunt het nu als volgt activeren:

```
B50 PROGS
```

Er wordt een backup gemaakt van alle bestanden in de directory PROGS en de subdirectories van die directory met de bijbehorende protocolbestanden. Overal waar in het batchbestand de parameter %1 staat, wordt die vervangen door de naam van de directory, PROGS.

In de volgende paragraaf gaan we dieper in op batch-opdrachten en de bijzonderheden van batchbestanden.

1.3 De principes van een batchbestand

We hebben deze paragraaf in stukjes verdeeld onder tussenkoppen, zodat u kunt uitzoeken wat van uw gading is. Een aantal zaken die in de vorige paragraaf voor beginners uitgebreid zijn beschreven, komen in deze paragraaf slechts summier aan de orde.

Wat zijn batchbestanden?

Batchbestanden zijn bestanden met de extensie .BAT, die bestaan uit een verzameling DOS-opdrachten, die achtereenvolgens worden uitgevoerd. Het meest gebruikte batchbestand is het bestand AUTOEXEC.BAT, dat direct na de start van het systeem wordt uitgevoerd.

Batchbestanden maken

U kunt batchbestanden op verschillende manieren aanmaken, zolang het bestand maar wordt opgeslagen in ASCII-code. Iedere opdrachtregel moet worden afgesloten met Enter. Kleine, overzichtelijke batchbestanden kunt u maken met de DOS-opdracht COPY CON:

```
COPY CON NAAM.BAT
```

Het bestand wordt opgeslagen met behulp van Ctrl-Z of F6 onder de naam NAAM.BAT. U kunt alleen fouten verbeteren in de actuele regel. Ontdekt u een fout in een met Enter afgesloten regel, dan zult u helemaal overnieuw moeten beginnen of het bestand moeten opslaan en daarna laden in een tekstverwerker.

Vanwege die beperking verdient de editor EDIT de voorkeur. In tegenstelling tot COPY CON, een interne DOS-opdracht, is EDIT een externe DOS-opdracht. Dat houdt in dat het programma van schijf moet worden gelezen voordat het ter beschikking staat. Roep het programma op met:

```
EDIT NAAM.BAT
```

EDIT is in feite een doorgeefluik: u roept er de editor van QBASIC mee op. Voer de opdrachten in en sluit iedere regel af met Enter. Sla het bestand op met de functie Save van het menu File.

U kunt ook batchbestanden maken met een tekstverwerker, mits die het bestand kan opslaan in ASCII-code. Het voordeel is dat u in een vertrouwde omgeving werkt, namelijk uw eigen tekstverwerker; het nadeel is dat u eerst de tekstverwerker moet laden, wat een tijdrovende aangelegenheid is, zeker als u zonder harde schijf werkt.

Batchbestanden starten

U start een batchbestand net zoals andere DOS-opdrachten, door de naam van het bestand in te voeren. De extensie .BAT laat u daarbij achterwege. Dat weglaten van de extensie geeft aanleiding tot een paar opmerkingen.

Rangorde van bestanden met de extensies .COM, .EXE en .BAT

Batchbestanden zijn niet de enige DOS-opdrachten waarbij u de extensie mag weglaten; dat geldt ook voor programma's met de extensie .COM of .EXE, program-

ma's dus, die DOS direct kan uitvoeren. Om verwarring te voorkomen zijn de drie soorten DOS-opdrachten in een hiërarchie ondergebracht.

Veronderstel dat een directory de programma's TEST.COM en TEST.EXE bevat. U maakt zelf een batchbestand TEST.BAT aan en slaat dat op in dezelfde directory. Het besturingssysteem trekt zich nu niets aan van uw poging om met de volgende opdracht het batchbestand te activeren:

```
TEST
```

Programma's met de extensie .COM krijgen namelijk de hoogste prioriteit en in dit geval zal TEST.COM dus worden uitgevoerd. Zou u TEST.COM wissen, dan gaat TEST.EXE nog voor TEST.BAT. Het verdient dus aanbeveling het batchbestand een afwijkende naam te geven of onder te brengen in een andere directory. We komen daar later nog op terug.

Zoekpaden gebruiken

De opdracht PATH (pad) speelt een grote rol in MS-DOS en derhalve ook in dit boek. Achter PATH geeft u de namen op van drives en directories waar DOS bestanden moet zoeken.

Als u geen pad specificeert, controleert DOS eerst of de opdracht wellicht een interne opdracht is zoals DIR of DATE. Is dat het geval, dan wordt die opdracht uitgevoerd. Betreft het geen interne opdracht, dan zoekt DOS in dit geval alleen in de actuele directory naar de programmanaam waaruit de opdracht bestaat.

Om te bereiken dat MS-DOS uit uw batchbestanden haalt wat er in zit, kunt u ze het beste allemaal onderbrengen in een aparte directory, bijvoorbeeld de directory C:\BAT, en een zoekpad specificeren naar die directory. Neem de opdracht PATH op in uw bestand AUTOEXEC.BAT, dan kunt u vanaf het moment dat de computer aan staat, gebruik maken van uw batchbestanden.

U specificeert een pad door de opdracht PATH in te voeren achter de DOS-prompt. De specificatie vervangt dan oudere specificaties. U kunt ook een regel toevoegen aan uw bestand AUTOEXEC.BAT; de opdracht ziet er als volgt uit:

```
PATH C:\BAT
```

Meerdere zoekpaden in één keer opgeven is ook toegestaan. U dient de zoekpaden dan te scheiden door een puntkomma:

```
PATH C:\BAT;C:\DOS;C:\WP
```

In het bovenstaande voorbeeld wordt eerst de actuele directory doorzocht, dan de directory BAT, daarna de directory DOS en ten slotte de directory WP. Geef 'C:\BAT' op als eerste zoekpad, dat biedt u de mogelijkheid externe DOS-opdrachten te vervangen door zelfgemaakte batchbestanden.

Het is goed om altijd in het achterhoofd te houden dat het niet functioneren van een voorbeeldprogramma heel goed kan worden veroorzaakt door een onjuist zoekpad.

Voert u de opdracht PATH in zonder specificatie, dan toont DOS u de actuele zoekpaden. Wilt u daar een nieuw zoekpad aan toevoegen, dan zult u alle bestaande zoekpaden eerst weer moeten typen, gevolgd door een puntkomma en het nieuwe zoekpad. Dat dat ook sneller kan, kunt u verderop in dit boek lezen in de Quicktip onder de naam ADDPATH.

Bij de volgorde waarin u uw specificaties van zoekpaden koppelt aan PATH heeft het zin de rangorde .COM, .EXE en .BAT in de gaten te houden. Omdat de zoekpaden in de eerder vermelde volgorde worden doorzocht, maakt het batchbestand TEST.BAT nu wel een kans ten opzichte van het bestand TEST.COM.

1.4 Batchbestanden programmeren

In deze paragraaf willen we u aan de hand van voorbeelden vertrouwd maken met de opdrachten die u kunt gebruiken in batchbestanden.

De opdracht ECHO

De opdracht ECHO heeft twee functies. Voegt u er een tekst aan toe, dan wordt die uitgevoerd naar het actuele uitvoerapparaat (meestal het scherm). U kunt daarvan gebruik maken om allerlei meldingen uit te voeren naar het scherm.

U kunt de opdracht ook koppelen aan de parameters ON en OFF. Daarmee geeft u aan of de volgende opdrachten van het batchbestand wel of niet moeten worden weergegeven op het scherm. Een batchbestand vervangt weliswaar de handmatig ingevoerde opdrachten, maar zet normaal gesproken alle opdrachten wel op het scherm. U kunt die weergave onderdrukken door de opdracht ECHO OFF aan het begin van het batchbestand op te nemen. Met ECHO ON herstelt u de oorspronkelijke toestand weer. Met de opdracht ECHO zonder parameters vraagt u de actuele toestand op.

In het voorbeeld gaan we uit van een batchbestand TEKSTEN, dat de directory activeert waarin u uw tekstbestanden voor de tekstverwerker opslaat. Aansluitend verzorgt het batchbestand een overzicht van al die tekstbestanden per (scherm)pagina. Het batchbestand bevat in principe de volgende opdrachten:

```
CD C:\TEKSTEN
DIR /P
```

Laat u het daarbij, dan werkt het bestand uitstekend, maar worden ook de opdrachten zelf op het scherm weergegeven. Voor het functioneren van de opdrachten is dat niet nodig, terwijl het beeld er hinderlijk onrustig door wordt. Breidt u daarom het batchbestand uit met de ECHO-opdrachten:

```
@ECHO OFF
CD C:\TEKSTEN
DIR /P
ECHO ON
```

Beide opdrachten waar het om gaat, worden nu niet op het scherm weergegeven. De apestaart (@) zorgt ervoor dat de opdracht ECHO OFF eigenlijk met terugwerkende kracht ook slaat op die opdracht zelf, anders zou die nog wel op het scherm worden weergegeven. De apestaart leent zich goed om een enkele opdracht uit een batchbestand te onderdrukken.

Gebruik de opdracht ECHO ON aan het eind van het batchbestand om er zeker van te zijn dat de prompt op uw scherm verschijnt. Er zijn omstandigheden waarin dat niet gebeurt na ECHO OFF. We breiden het batchbestand nog uit met een regel waarin de andere functie van de opdracht ECHO aan bod komt:

```
@ECHO OFF
ECHO Het batchbestand activeert de directory met tekstbestanden
ECHO en geeft een overzicht van de inhoud.
CD C:\TEKSTEN
DIR /P
@ECHO ON
```

Als u het batchbestand activeert, worden de tweede en de derde regel op uw scherm weergegeven. U dient de opdracht ECHO te herhalen op iedere regel die moet worden uitgevoerd. Bevat de directory met tekstbestanden meer bestanden dan er op het scherm passen, dan moet u wel snel kijken, want de boodschap scrollt in dat geval boven uit het beeld.

ECHO is dus een opdracht waarmee u op een handige manier informatie naar het scherm kunt sturen. Dat dat erg belangrijk kan zijn, zullen we u verderop laten zien. In het onderhavige voorbeeld dient de opdracht alleen ter verduidelijking van wat zich achter de schermen afspeelt.

De opdracht REM

REM is een afkorting van 'remark' (opmerking) en biedt de mogelijkheid commentaar op te nemen in een batchbestand. Het commentaar wordt door DOS genegeerd, maar maakt lange batchbestanden veel leesbaarder.

In dergelijke lange bestanden kunt u bijvoorbeeld sub-programma's van het batchbestand markeren met commentaar, zodat het redigeren en wijzigen van een bestand makkelijker wordt.

In het voorbeeld maken we een batchbestand dat op het scherm een overzicht geeft van de bestanden in de actuele directory, gesorteerd op extensie. We geven het bestand de naam EXTSORT.BAT en voegen commentaar toe om het bestand later snel te kunnen herkennen.

```
@ECHO OFF
REM ************************************
REM *      Overzicht van de inhoud     *
REM *          van een directory       *
REM *      gesorteerd op extensie      *
REM ************************************
DIR /O:E /P
@ECHO ON
```

Activeert u het batchbestand, dan worden de commentaarregels niet op het scherm weergegeven. Pas als u het batchbestand in een editor laadt of met de opdracht TYPE op het scherm zet, wordt ook het commentaar weer zichtbaar.

Batchbestanden afbreken

DOS-opdrachten kunt u afbreken met Ctrl-C of Ctrl-Break. Onderbreekt u de uitvoering van de opdrachten van een batchbestand met een van beide toetsencombinaties, dan verschijnt de volgende vraag op uw scherm:

```
Terminate batch job (Y/N)?
```

Afhankelijk van uw antwoord wordt de uitvoering van de batch-opdrachten voortgezet of afgebroken.

Tip: U kunt de uitvoering van een DOS-opdracht of een batchbestand alleen afbreken als het systeem daarop is ingesteld. Als dat niet het geval is, kunt u alleen onder bepaalde omstandigheden de uitvoering afbreken, wat soms erg vervelend kan zijn.

```
BREAK = ON
```

De bovenstaande opdracht behoort om die reden een van de eerste opdrachten te zijn in uw AUTOEXEC.BAT

Hoe het afbreken in zijn werk gaat, kunt u uittesten aan de hand van het hierboven beschreven batchbestand EXTSORT. Activeer het bestand en druk vervolgens op Ctrl-C. Op enig moment tijdens de uitvoering van de opdrachten verschijnt dan de controlevraag op uw scherm en kunt u besluiten hoe het verder moet.

De opdracht PAUSE

De opdracht PAUSE onderbreekt de uitvoering van de batch-opdrachten en wacht op een toetsaanslag. Nadat u op een willekeurige toets hebt gedrukt, gaat het programma verder. Dit geeft u de mogelijkheid het programma te onderbreken met Ctrl-C.

We zullen u de mogelijkheden van de opdracht laten zien aan de hand van het volgende voorbeeld. Het batchbestand in kwestie wist alle bestanden met de extensie .BAT. Dat is een gevaarlijke opdracht en dus is een controlevraag geboden. Voer de volgende tekst in:

```
@ECHO OFF
ECHO Alle bestanden met de extensie .BAT worden gewist!
ECHO Druk op Ctrl-C om af te breken
PAUSE
DEL *.BAT
@ECHO ON
```

Als u nu het batchbestand start, worden de beide regels met informatie op uw scherm geplaatst. Daarna wacht het systeem tot u op een toets drukt. Drukt u op Ctrl-C, dan wordt het batchbestand afgebroken en worden de batchbestanden niet gewist. Drukt u op een willekeurige andere toets, dan worden de batchbestanden van de actuele directory gewist.

Parameters doorgeven

Batchbestanden zijn erg flexibel; Terwijl u ze aanmaakt, kunt u ze heel algemeen houden, waardoor het gebruik van parameters mogelijk wordt.

De variabelen in het batchbestand worden aangegeven met het procentteken en een getal tussen nul en tien. In één batchbestand kunt u dus zonder meer negen parameters gebruiken.

Bij het starten van het batchbestand plaatst u de parameter op de gebruikelijke manier achter de DOS-opdracht (hier de naam van het batchbestand), gescheiden door een spatie.

We lichten het gebruik van parameters toe aan de hand van een batchbestand DELSUB.BAT, dat een subdirectory wist. De directory moet leeg zijn; dus moeten eerst de bestanden daarin worden gewist. Daarna wordt de directory zelf gewist.

We zullen het batchbestand eerst aanmaken, alsof het alleen voor deze ene directory wordt gebruikt. We noemen de directory TEMP. Als die nog niet op uw harde schijf voorkomt, maakt u hem als volgt:

```
MD C:\TEMP
```

We kunnen nu de opdrachten voor het batchbestand invoeren:

```
@ECHO OFF
REM De subdirectory TEMP wissen
DEL C:\TEMP
RD C:\TEMP
ECHO ON
```

De eerste opdracht die wordt uitgevoerd, luidt:

```
DEL C:\TEMP
```

Dat is een toegestane verkorte vorm van de opdracht:

```
DEL C:\TEMP\*.*
```

Voordat die opdracht wordt uitgevoerd, dient u de controlevraag met Yes te beantwoorden. De volgende opdracht wist dan de (lege) directory:

```
RD C:\TEMP
```

Tip: Als u het batchbestand start, terwijl de directory niet bestaat, volgt een foutboodschap.

Zouden er aan de directory TEMP nog subdirectories zijn gekoppeld, dan volgt ook een foutboodschap omdat TEMP niet leeg is. De opdracht RD (Remove Directory) kan dan niet worden uitgevoerd.

We zullen dit batchbestand algemeen bruikbaar maken, zodat elke directory ermee kan worden gewist. Daarvoor maken we gebruik van een variabele in het batchbestand die wordt ingevuld door een parameter bij de start van het batchbestand.

De parameter is in dit geval de naam van de te wissen directory. De opdracht waarmee TEMP wordt gewist, luidt dan:

```
DELSUB C:\TEMP
```

Zolang het batchbestand niet is gewijzigd, heeft de parameter geen zin. De directory TEMP zou toch wel worden gewist. Het batchbestand moet zodanig worden aangepast dat het systeem gedwongen wordt de parameter te lezen. Daarom vervangen we in het batchbestand overal de beschrijving van de directory C:\TEMP door %1. Zouden we nog een tweede variabele willen inbouwen, dan werd dat %2; een derde werd %3, enzovoort. In principe kunt u maximaal negen variabelen gebruiken.

MS-DOS vat de parameters op als strings. Om onderscheid te kunnen maken tussen meerdere parameters moeten ze telkens van elkaar en van de opdracht worden gescheiden door een spatie.

Het batchbestand komt er als volgt uit te zien:

```
@ECHO OFF
REM Een willekeurige subdirectory wissen
DEL %1
RD %1
ECHO ON
```

Als u nu het batchbestand start met de opdracht:

```
DELSUB C:\TEMP
```

wordt de parameter 'C:\TEMP' gebruikt om de variabele %1 inhoud te geven. Eerst

worden de bestanden gewist en vervolgens de directory zelf.

Tip: Voordat u een directory wist, kunt u beter nog even controleren of u dat wel werkelijk wilt. Vraag met DIR eerst nog een keer een overzicht van de bestanden in de directory op. Vergissingen kunnen erg pijnlijk zijn.

Tip: Wilt u zien hoe MS-DOS de parameter gebruikt, wis dan de eerste regel van uw batchbestand of vervang OFF door ON.

In het volgende voorbeeldbestand, DELDAT.BAT, gebruiken we twee variabelen, zodat u bij de opdracht om het bestand te starten ook twee parameters zult moeten meegeven. Het doel van het bestand is het wissen van bepaalde bestanden in een directory naar keuze. Voordat de bestanden worden gewist, verschijnen ze nog in een overzichtje van op het scherm en krijgt u de mogelijkheid het proces met Ctrl-C af te breken. Het batchbestand ziet er als volgt uit:

```
@ECHO OFF
REM Bestanden naar keuze wissen in een directory naar keuze
CD %1
DIR %2
ECHO De bovenstaande bestanden worden gewist
ECHO tenzij u op Ctrl-C drukt
PAUSE
DEL %2
ECHO ON
```

Let op: Vergewis u ervan dat de aangegeven directory bestaat! Als die namelijk niet bestaat, worden de bestanden in de actuele directory gewist. Dat is ook het belang van de opdracht 'DIR %2': u krijgt de bestanden nog een keer te zien en kunt ingrijpen.

In de praktijk gebeurt het volgende. Veronderstel dat u het batchbestand als volgt oproept:

```
DELDAT C:\TEMP *.TXT
```

De parameter C:\TEMP vormt nu de inhoud van de variabele %1 en *.TXT vult de variabele %2. Het eerste wat gebeurt, is dus het activeren van de directory %1, oftewel C:\TEMP. Daarna wordt een overzicht gegeven van de bestanden %2, oftewel alle bestanden met de extensie .TXT. U kunt daarna zelf bepalen of u verder wilt gaan of niet.

U kunt hetzelfde resultaat op meerdere manieren bereiken, maar het doel van dit voorbeeldbestand was u het werken meer dan één parameter te laten zien. Het belang van die parameters zal hopelijk duidelijk worden bij de bespreking van de volgende opdrachten.

De opdracht FOR ... DO

Met de opdracht FOR ... DO kunt u lussen in uw batchbestand inbouwen. Een lus is een programmadeel dat een aantal malen achter elkaar wordt afgewerkt. De syntaxis van deze opdracht is wat ingewikkeld en behoeft nadere toelichting.

We doen dat aan de hand van een klein batchbestand GETAL.BAT. Het bestand toont achtereenvolgens de getallen één tot en met vier:

```
@ECHO OFF
FOR %%a IN (1 2 3 4) DO ECHO Dit is een %%a
ECHO ON
```

In dit bestand is een FOR ...DO-lus ingebouwd. %%a staat voor een variabele, die achtereenvolgens de waarden aanneemt van de eropvolgende lijst tussen haakjes. In het voorbeeld neemt %%a de waarden één, twee, drie en vier aan. De waarden in de lijst tussen haakjes moeten van elkaar worden gescheiden door spaties, komma's of puntkomma's. Steeds als de variabele een nieuwe waarde heeft gekregen, wordt de erachterstaande opdracht uitgevoerd. In het voorbeeld dus steeds de opdracht om met behulp van ECHO een boodschap op het scherm te zetten, waarin dezelfde variabele is verwerkt.

De opdracht biedt u dus de mogelijkheid een andere opdracht met verschillende waarden uit te voeren.

De waarden in de lijst kunt u natuurlijk ook vervangen door bestandsnamen. Dat maakt het mogelijk een opdracht uit te voeren voor verschillende bestanden. Zo zou u in één keer de inhoud van meerdere bestanden op het scherm kunnen weergeven met de opdracht TYPE:

```
FOR %%a IN (AUTOEXEC.BAT CONFIG.SYS) DO TYPE %%a
```

Als beide bestanden in de actuele directory staan, wordt de inhoud ervan op het scherm weergegeven.

Wildcards gebruiken in lussen

Veronderstel dat u in een directory met 30 tekstbestanden een bepaald tekstbestand zoekt. U zou dan een lijst van 30 bestandsnamen moeten opnemen in de hierboven beschreven opdracht. Vermoeiend, vervelend en het schreeuwt om fouten. Gelukkig kan het eenvoudiger: u kunt in de lijst wildcards gebruiken. De volgende opdracht zet de inhoud van alle bestanden met de extensie .TXT van de actuele directory op uw scherm:

```
FOR %%a IN (*.TXT) DO TYPE %%a
```

De syntaxis van de opdracht ziet er uiteindelijk dus als volgt uit:

```
FOR %%V IN (Lijst) DO OPDRACHT %%V
```

%%V De variabele V neemt achtereenvolgens alle waarden aan uit de lijst. Als u de opdracht rechtstreeks achter de prompt invoert, kunt u volstaan met één procentteken; in batchbestanden moet u twee procenttekens gebruiken.

Lijst De lijst bevat de elementen waarop de opdracht van toepassing is. Hij moet altijd tussen haakjes staan. De lijst mag bestaan uit een opsomming (bestand1 bestand2 bestand3), maar u mag ook wildcards gebruiken (*.TXT). In het eerste geval neemt de variabele V achtereenvolgens de waarden bestand1, bestand2 en bestand3 aan; in het tweede geval wordt de variabele een niet van tevoren bepaald aantal malen gevuld met de naam van een tekstbestand.

Opdracht Na DO geeft u een DOS-opdracht, die bij iedere nieuwe waarde wordt uitgevoerd.

De opdracht GOTO

Tot nu toe hebben we alleen gewerkt met batchbestanden waarin de opdrachten keurig netjes op volgorde werden afgewerkt. Er zijn echter veel situaties waarin een sprong naar een andere opdracht erg handig is.

Daarvoor maken we gebruik van een *label* en geven we DOS met GOTO de opdracht bij de label verder te gaan met de uitvoering van opdrachten.

De label moet aan het begin van een regel staan en wordt voorafgegaan door een

dubbele punt. De eerste acht tekens van de label worden door DOS gebruikt. Is de label langer, dan worden de overige tekens genegeerd. Een opdracht om naar een andere plaats te springen bestaat dus altijd uit de volgende combinatie:

```
GOTO label
...
:label
```

Het volgende voorbeeld SPRONG.BAT heeft in de praktijk weinig betekenis, maar laat goed zien wat de bedoeling is van de opdracht:

```
@ECHO OFF
ECHO Er volgt nu een sprong
GOTO label
ECHO Deze boodschap verschijnt nooit op het scherm
:label
ECHO DOS is naar deze plaats gesprongen
@ECHO ON
```

Als de boodschap uit de vierde regel van het programma toch op uw scherm verschijnt, is er iets fout gegaan. DOS hoort die regel over te slaan en naar de label 'label' te springen.

De opdracht IF

Met IF kunt u voorwaarden controleren. Wordt aan de voorwaarde voldaan, dan wordt de rest van de regel afgewerkt; zo niet, dan gaat DOS verder met de opdracht op de volgende regel.

Met behulp van een dubbel gelijkteken kunt u controleren of twee parameters identiek zijn. We lichten dat toe aan de hand van een batchbestand TEST.BAT. Het bestand controleert of het batchbestand is gestart met de parameter 'Help' en onderneemt dienovereenkomstig actie. De opdracht ziet er als volgt uit:

```
IF %1==Help ECHO De parameter Help is opgegeven
```

De opdracht IF verwacht zowel voor als achter het dubbele gelijkteken invoer. Zou u geen parameter invoeren, dan zou dat tot een foutmelding leiden. De opdracht wordt dan door DOS namelijk op de volgende manier geïnterpreteerd:

```
IF ==Help ECHO De parameter Help is opgegeven
```

Omdat er nu duidelijk iets ontbreekt voor het gelijkteken, wat niet is toegestaan, dient u domweg aan beide zijden van de vergelijking iets toe te voegen. Dubbele aanhalingstekens lenen zich daar goed voor.

```
IF "%1"=="Help" ECHO De parameter Help is opgegeven
```

Als u nu het batchbestand oproept, zal de boodschap alleen op het scherm verschijnen als de parameter Help is toegevoegd:

```
TEST Help
```

Tip: Bij de vergelijking wordt onderscheid gemaakt tussen hoofdletters en kleine letters. De volgende opdracht leidt ertoe dat er geen boodschap op het scherm wordt weergegeven:

```
TEST HELP
```

In dit soort gevallen is 'HELP' voor DOS iets anders dan 'Help'.

U kunt IF koppelen aan meer voorwaarden:

IF NOT voorwaarde — DOS controleert of aan een voorwaarde niet wordt voldaan en werkt dan de rest van de regel af. Is wel aan de voorwaarde voldaan, dan gaat DOS verder met de opdracht van de volgende regel.

```
IF NOT "%1"=="Help" ECHO De parameter Help is NIET opgegeven
```

IF EXIST bestand — DOS controleert of het opgegeven bestand bestaat. Zo ja, dan wordt de rest van de regel afgewerkt; zo niet, dan gaat DOS verder met de opdracht van de volgende regel. U kunt deze opdracht ook weer negatief gebruiken in de vorm IF NOT EXIST. De rest van de regel wordt dan alleen afgewerkt, als het bestand niet bestaat.

```
IF EXIST \AUTOEXEC.BAT COPY AUTOEXEC.BAT AUTOEXEC.BAK
```

IF ERRORLEVEL — DOS onderzoekt foutcodes (errorlevels). Een aantal DOS-opdrachten geeft bij een fout of een probleem een foutcode terug aan het batchbestand. Met ERRORLEVEL kunt u de waarde van de foutcode opvragen. Als er geen fout is opgetreden is de errorlevel altijd nul. Als errorlevel gelijk is aan

of groter dan de opgegeven waarde, voert DOS de volgende regel uit:

```
FORMAT A:
IF ERRORLEVEL 3 ECHO Format is afgebroken met Ctrl-C
```

Met behulp van de opdracht IF kunt u de loop van het batchbestand dus sturen. Als u de opdracht koppelt aan GOTO, kunt u al naar gelang een sprong naar een andere plaats in het batchbestand maken.

In het voorbeeld dat nu volgt, bouwen we een routine in om de geldigheid van opgegeven parameters te onderzoeken. Als de gebruiker een fout maakt, wordt een boodschap met de juiste syntaxis op het scherm gezet. We noemen het batchbestand TYPEM.BAT (TYPE MORE). Het batchbestand controleert of een bepaald bestand aanwezig is en geeft het per scherm weer. Als er iets niet klopt, wordt een aanwijzing op het scherm weergegeven.

```
@REM ****************************************************
@REM * Dit batchbestand geeft een bestand per scherm weer *
@REM ****************************************************
@ECHO OFF
IF "%1" == "" GOTO Syntaxis
IF NOT EXIST %1 GOTO Niet_gevonden
TYPE %1 | MORE
GOTO Einde
:Syntaxis
ECHO Dit batchbestand geeft een bestand per scherm weer
ECHO Syntaxis: TYPEM BESTAND.EXT
GOTO Einde
:Niet_gevonden
ECHO Het bestand is niet gevonden
GOTO Einde
:Einde
ECHO ON
```

Het voorbeeldprogramma onderzoekt twee mogelijke fouten:

1 Als u het programma oproept zonder parameter, springt het naar de routine Syntaxis. Daar wordt de bedoeling van het bestand beschreven en een aanwijzing om een bestandsnaam in te voeren als parameter.

Tip: Zie hoofdstuk 2 als u meer informatie wilt over de sprongopdracht naar

de label Syntaxis. Daar wordt de tip 'Zo herkent u ontbrekende parameters' besproken.

2 Als u wel een bestandsnaam invoert, maar het systeem het bestand niet kan vinden, springt het programma naar 'Niet_gevonden' en voert het een melding uit naar het scherm.

We lichten de werking van IF en GOTO nogmaals toe, nu aan de hand van een voorbeeld waarin het aantal parameters van belang is voor het vervolg.

Als u in een batchbestand alle parameters wilt verwerken, kunt u ze uiteraard met de bijbehorende opdrachten onder elkaar invoeren. Zo kunt u een batchbestand maken dat alle als parameters opgegeven bestanden weergeeft op het scherm. Het batchbestand DATAN.BAT ziet er als volgt uit.

```
@REM ******************************************************
@REM * Dit batchbestand geeft de opgegeven bestanden weer *
@REM ******************************************************
@ECHO OFF
IF "%1" == "" GOTO Einde
TYPE %1
PAUSE
IF "%2" == "" GOTO Einde
TYPE %2
PAUSE
IF "%3" == "" GOTO Einde
TYPE %3
:Einde
@ECHO ON
```

Dit is een computer-onvriendelijk batchbestand: dezelfde reeks opdrachten komt drie keer voor, waarbij het eigenlijk steeds om dezelfde variabele gaat. Bovendien is het met dit batchbestand niet mogelijk meer dan drie bestanden weer te geven. De opdracht SHIFT doet daar wat aan.

De opdracht SHIFT

Hiervoor hebben we al aangegeven dat u in principe negen parameters mag gebruiken in een batchbestand. In het batchbestand vervangen ze genummerde variabelen, voorafgegaan door een procentteken. Meer parameters kunt u niet in één keer in het batchbestand verwerken. De opdracht SHIFT koppelt alle parameters aan een varia-

bele die een nummer lager ligt. De inhoud van %9 wordt verplaatst naar %8, de inhoud van %8 naar %7, enzovoort. De inhoud van %0, de naam van het batchbestand, gaat verloren; een tiende parameter wordt gekoppeld aan de variabele %9.

We laten dit zien aan de hand van een laatste voorbeeld, TYPEDAT.BAT.

```
@REM ****************************************************
@REM * Dit batchbestand geeft de opgegeven bestanden weer *
@REM ****************************************************
@ECHO OFF
:Verder
IF "%1" == "" GOTO Einde
TYPE %1
PAUSE
SHIFT
GOTO Verder
:Einde
@ECHO ON
```

De truc is dus dat steeds variabele %1 wordt verwerkt in een lus. De inhoud van de variabele is steeds anders, omdat SHIFT er een andere parameter aan koppelt. Het bestand controleert steeds of de parameter leeg is; zodra dat het geval is, springt het bestand naar het einde.

Batchbestanden oproepen vanuit batchbestanden

MS-DOS laat u meer batchbestanden aan elkaar koppelen. Dat leidt tot ongekende mogelijkheden op het terrein van batch-programmering. De twee manieren waarop u batchbestanden kunt koppelen zullen we kort toelichten aan de hand van voorbeelden.

De opdracht CALL

We hebben u bij de bespreking van de opdracht FOR ... DO laten zien hoe u variabelen gebruikt in een opdracht. In een FOR ... DO-lus kunt u echter slechts één opdracht een aantal malen herhalen. Wilt u meerdere opdrachten met de variabelen verbinden, dan wordt het moeilijker.

Plaats de opdrachten in een apart batchbestand en roep die op na de opdracht DO van de lus met de opdracht CALL. We laten zien hoe het werkt aan de hand van het batchbestand GEEFWEER.BAT.

De volgende opdracht geeft de inhoud van de bestanden AUTOEXEC.BAT en CONFIG.SYS weer op uw scherm:

```
FOR %%a IN (AUTOEXEC.BAT CONFIG.SYS) DO TYPE %%a
```

Wilt u de bestanden ook kopiëren naar een diskette, dan zult u opdrachten moeten toevoegen. Vervang daartoe de opdracht TYPE door CALL en maak een nieuw batchbestand, OPDRACHT.BAT, met de uit te voeren opdrachten:

```
FOR %%a IN (AUTOEXEC.BAT CONFIG.SYS) DO CALL Opdracht %%a
```

OPDRACHT.BAT ziet er als volgt uit:

```
TYPE %1
COPY %1 A:
```

Het eerste batchbestand roept het tweede batchbestand op met AUTOEXEC.BAT als parameter. Als de opdrachten van het tweede batchbestand zijn uitgevoerd, keert het systeem terug naar het eerste batchbestand en begint aan de tweede ronde langs de lus. OPDRACHT.BAT wordt nogmaals opgeroepen, nu met CONFIG.SYS als parameter.

De opdracht CALL zorgt dus voor de oproep van een tweede batchbestand, maar heeft altijd tot gevolg dat het systeem terugkeert naar het eerste batchbestand en daar de draad weer oppakt bij de volgende opdracht.

Batchbestanden koppelen zonder CALL

U kunt ook vanuit een batchbestand een tweede batchbestand oproepen zonder daarna terug te keren naar het eerste batchbestand. Dat kan zinvol zijn als alleen onder bepaalde voorwaarden bepaalde opdrachten moeten worden uitgevoerd. Die voorwaarden kunt u toetsen met de opdracht IF.

Fouten vermijden: ongeldige tekens in commentaar en ECHO-regels

Als u al eens na het starten van een zelfgeschreven batchbestand merkwaardige dingen zag gebeuren, is dat wellicht veroorzaakt door het gebruik van ongeldige tekens. De hieronder opgesomde tekens mag u namelijk niet zo maar overal gebruiken.

% DOS gebruikt het procentteken in combinatie met de cijfers één tot en met negen als plaatshouders voor parameters. Wilt het teken toch in lopende tekst gebruiken, bijvoorbeeld bij de opdracht ECHO, dan moet u het teken verdubbelen (%%).

< DOS gebruikt het kleiner-dan teken om de invoer van het toetsenbord te vervangen door invoer van een ander invoerapparaat of een bestand. Om die reden mag u het teken niet gebruiken in een batchbestand, ook niet in combinatie met de opdracht ECHO.

> Het groter-dan teken stuurt de uitvoer niet naar het scherm, maar naar een ander apparaat, bijvoorbeeld de printer of een bestand. U mag het teken evenmin gebruiken als het kleiner-dan teken.

| Het pipe-symbool leidt de uitvoer van een programma naar een ander programma (een zogenaamd filter). U mag het teken niet gebruiken in batchbestanden.

Wilt u een van deze ongeldige tekens toch per se op het scherm weergeven, dan schrijft u de uitvoer naar het scherm in een apart tekstbestand, dat u met de opdracht COPY naar het scherm kopieert. De volgende regel plaatst u met COPY CON in een apart tekstbestandje VERBODEN.DAT:

```
De tekens '< > |' zijn niet geldig in batchbestanden
```

Daarna maakt u een batchbestand VERBODEN.BAT met de opdracht:

```
COPY VERBODEN.DAT CON
```

De reden voor deze merkwaardige gang van zaken is dat MS-DOS de functie die aan deze tekens is gekoppeld altijd uitvoert, ook als ze deel uitmaken van uitvoer naar het scherm of commentaar.
Met het pipe-symbool kunt u nog een leuk grapje uithalen. Plaats in het batchbestand TEST.BAT de volgende opdracht:

```
REM Dit is een | test
```

Start het batchbestand. Niet alleen wordt de commentaarregel op uw scherm gezet, maar omdat na het pipe-symbool het woord 'test' wordt opgevat als de naam van het bestand waar de uitvoer naar toe moet worden geleid, wordt het programma opnieuw gestart om die uitvoer te ontvangen, en nog een keer, en nog een keer, net zo lang tot u op Ctrl-C drukt. DOS vraagt u dan op de bekende manier met 'Terminate batch job (Y/N)?' of u werkelijk wilt afbreken. Druk op 'Y' om te stoppen.

Tip: U kunt uiteraard deze ongeldige tekens wel op hun eigenlijke manier gebruiken in batchbestanden, dus om in- en uitvoer om te leiden.

U hebt nu de principes van het programmeren van batchbestanden onder de knie. U kunt de opdrachten op allerlei manieren combineren, waardoor een compleet netwerk van batchbestanden tot de mogelijkheden behoort. In het verdere verloop van dit boek zullen we nog een aantal malen nieuwe opdrachten toelichten waarmee u verfijningen kunt aanbrengen.

1.5 Programmeren in BASIC

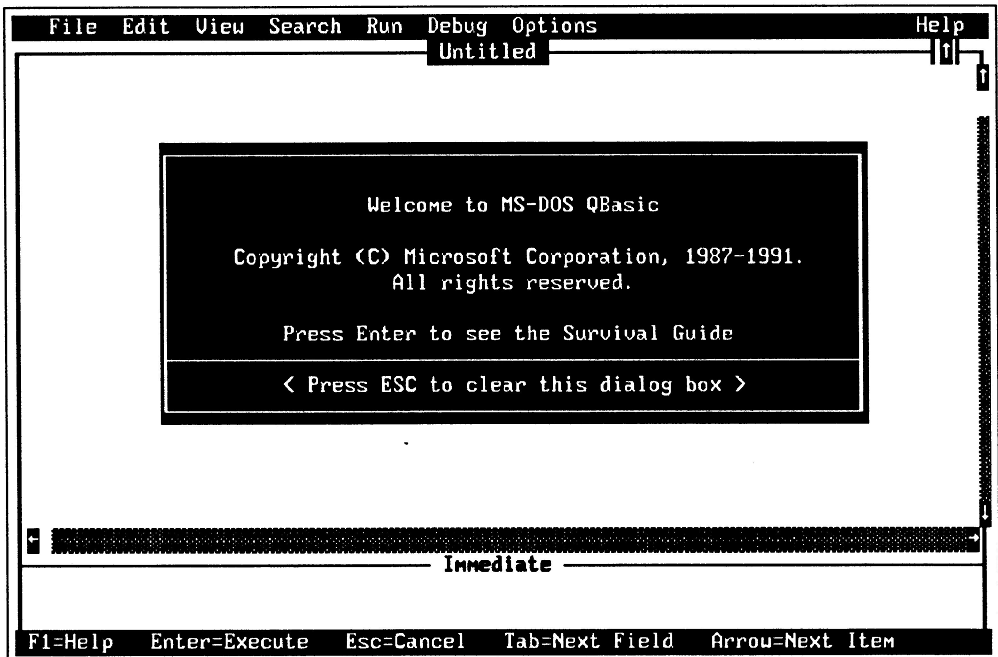
Niet alles wat we in dit boek willen laten zien, valt te bereiken met batchbestanden en DOS-opdrachten. Bovendien zijn sommige problemen makkelijker op een andere manier op te lossen. We maken daarom gebruik van de programmeertaal QBASIC, die door Microsoft wordt meegeleverd in het pakket van MS-DOS 5.0. We gaan ervan uit dat u nog niet eerder in QBASIC hebt geprogrammeerd en geven u eerst een overzicht van die zaken die u moet weten om QBASIC te kunnen gebruiken voor de tips en trucs in dit boek. U zult zien dat het schrijven van de programma's weinig moeite kost zonder dat dit ten koste gaat van hun prestaties.

QBASIC laden en starten

Om te beginnen dient u de bestanden van QBASIC op te zoeken op uw DOS-diskettes of uw harde schijf. Als u MS-DOS 5.0 op de aangegeven manier hebt geïnstalleerd, bevinden de bestanden zich in de directory \DOS. Zolang u nog geen speciaal batchbestand voor QBASIC hebt geschreven, laadt u het programma met de opdracht:

```
QBASIC
```

Op uw scherm ziet u nu het openingsscherm van QBASIC.



Het werkblad van QBASIC

Druk op Esc om het werkblad van QBASIC te activeren.

Programmaregels invoeren in QBASIC

Wilt u een QBASIC-programma uit dit boek gebruiken, dan neemt u eenvoudigweg de listing uit het boek over. Vanzelfsprekend mag u daarbij geen typefouten maken. Let u vooral op het gebruik van spaties en leestekens. Druk op Enter om een regel af te sluiten en de cursor naar een volgende regel te brengen.

Ontdekt u een typefout, dan brengt u de cursor met de cursortoetsen naar de bewuste plaats en corrigeert u de fout. De toets Backspace verwijdert het teken voor de cursor; de toets Del het teken boven de cursor. Met toets Ins schakelt u over tussen de invoegmodus en overschrijfmodus.

Ten behoeve van de overzichtelijkheid springen sommige regels in onze programma's in. Voor het functioneren van de programma's is dat niet van belang: u kunt eventueel alle regels langs de kantlijn laten beginnen.

QBASIC-programma's opslaan

Boven in het scherm bevindt zich de menuregel, die u activeert door op de Alt-toets te drukken. Drukt u vervolgens op de geaccentueerde letter, dan wordt het desbetreffende menu geopend. Met behulp van de geaccentueerde letters van de menu-opties kunt u een functie kiezen.

Als de menubalk actief is, is er ook steeds een keuze geaccentueerd met een markeringsbalk: de menunaam of de menu-optie wordt in een afwijkende kleur weergegeven. U kunt die keuze bevestigen door op Enter te drukken. In de pull-down menu's kunt u met de toetsen cursor omhoog en cursor omlaag andere opties markeren.

Sla een programma als volgt op. Activeer de menubalk door op Alt te drukken. Druk op Enter om het menu File te openen. Plaats de markering op Save As... met behulp van de toets cursor omlaag. Op uw scherm verschijnt een dialoogvenster waarin u een naam kunt opgeven voor het programma en een directory kunt uitkiezen waar het moet worden opgeslagen.

Gebruik in dialoogvensters altijd de Tab-toets om van het ene naar het andere invoerveld te springen of een keuzeschakelaar te activeren.

QBASIC-programma's laden en wijzigen

Laad een programma door nogmaals het menu File te openen. Kies Open... en voer de naam in van het programma. Pas eventueel eerst de directory aan. Bevestig uw keuze met Enter. Het dialoogvenster wordt gesloten en de listing van het programma verschijnt op het werkblad.

Als het programma te groot is voor het werkblad, gebruikt u de toetsen cursor omhoog en cursor omlaag om het scherm te scrollen.

Wijzigingen brengt u aan op de manier die we hierboven bij 'Programmaregels invoeren..' hebben beschreven. Daarna kunt u kiezen of u het programma onder dezelfde naam wilt opslaan of onder een andere naam. Dat laatste is vooral handig als u op grote schaal wijzigingen hebt aangebracht waarvan u niet zeker weet of ze juist zijn. Om het programma onder dezelfde naam op te slaan opent u het menu File en kiest u Save. QBASIC heeft de naam van uw programma onthouden en u hoeft verder niets te doen. Wilt u het programma onder een andere naam opslaan, dan kiest u voor Save As....

Programma's uitvoeren en onderbreken

Open het menu Run en kies Start om een programma te starten. Wilt u een programma dat bezig is, onderbreken, dan drukt u op Ctrl-C of Ctrl-Break.

In alle gevallen is het beter een zojuist geschreven programma eerst op te slaan, voordat u het start. Mocht het programma onoverkomelijke fouten bevatten, waardoor de computer blijft hangen, dan kunt u het na een reset van het systeem opnieuw laden in de editor van QBASIC; soms hoeft u maar één regeltje te wijzigen om de fout te corrigeren. Hebt u het programma niet opgeslagen, dan zult u het helemaal overnieuw moeten typen.

Reageren op foutmeldingen

Als u een programma start, kunnen er foutmeldingen komen. De meest voorkomende foutmelding is:

```
Syntax error
```

QBASIC zet de listing van uw programma op het werkblad en markeert de fout. Vergelijk de door u ingevoerde tekst met die in het boek. Veelal gaat het om kleine foutjes: een vergeten puntkomma of iets dergelijks. Corrigeer de fout, sla het programma op en start het opnieuw.

Er kunnen echter ook heel andere foutmeldingen op uw scherm verschijnen. Het voert te ver om daar in detail op in te gaan. Vergelijk in dat geval de hele programma-listing met die in het boek. In de meeste gevallen zult u dan toch de bewuste fout ontdekken.

QBASIC-programma's compileren

Het programma QBASIC dat Microsoft meelevert met MS-DOS 5.0 is een *interpreter*. Dat houdt in dat QBASIC nadat u een programma start, de regels stuk voor stuk langsgaat, de opdrachten controleert en dan pas uitvoert. De tegenhanger van een interpreter is een *compiler*. Het programma dat met een compiler wordt verwerkt, wordt niet onmiddellijk gestart. Het wordt eerst in zijn geheel gecontroleerd en klaar gemaakt en meestal in gecompileerde vorm opgeslagen. Pas daarna kan het worden gestart.

Beide manieren hebben hun voor- en nadelen. Op het eerste gezicht lijkt het dat een interpreter sneller is dan een compiler, maar het tegendeel is waar: het voortdurend controleren en vertalen van opdrachten kost veel rekentijd van de processor. Dat is te zien aan het volgende voorbeeld:

```
FOR I = 1 TO 1000
  PRINT I
NEXT I
```

In deze lus wordt de tweede regel 1000 maal gecontroleerd en vertaald in voor het systeem begrijpelijke code, namelijk steeds als QBASIC de opdracht moet uitvoeren. Dat is een duidelijk nadeel van de interpreter. Het voordeel van de interpreter is dat beginnende programmeurs direct toegang hebben tot het programma en makkelijk wijzigingen kunnen aanbrengen.

Een compiler vertaalt het programma in onbegrijpelijke tekens. De beginnende programmeur kan daar niet zo veel mee. Maar een gecompileerd programma werkt veel sneller: de opdracht 'PRINT I' van het bovenstaande voorbeeld wordt bij het compileren één keer gecontroleerd en vertaald, wat bij uitvoering een tijdwinst oplevert van zo'n 66 procent. Wel duurt het aanvankelijke controleren en vertalen van de code langer en die tijd heeft de compiler na iedere wijziging opnieuw nodig. Moderne compilers werken echter behoorlijk snel, zodat de voordelen groter zijn dan de nadelen.

Een ander voordeel van gecompileerde programma's is dat u het programma kunt starten zonder tussenkomst van QBASIC, dus rechtstreeks vanaf de DOS-opdrachtregel. Een gebruiker die alleen over een gecompileerde versie beschikt, kan die bovendien niet bekijken of wijzigen.

Mocht u na verloop van tijd plezier krijgen in het programmeren in QBASIC, dan bevelen we u de QBASIC-compiler aan. De programma's in dit boek draaien zonder uitzondering met behulp van de QBASIC-interpreter, maar bij veelvuldig gebruik wordt het nogal vervelend om ieder keer eerst de interpreter te moeten laden met het programma en daarna pas het programma te starten met Run en Start.

1.6 Machinetaal en Debug

Schrik niet, u bent niet plotseling verzeild geraakt in een boek voor systeemprogrammeurs. Voor sommige tips en trucs in dit boek zijn batchbestanden en QBASIC-programmaatjes echter niet toereikend. In die gevallen zullen we op een onopvallende manier machinetaal gebruiken.

Machinetaal (in het Engels *assembler*) is de moedertaal van uw computer. Om BASIC te begrijpen heeft het systeem nog altijd een tolk nodig; voor machinetaal geldt dat niet, uw computer kan de opdrachten rechtstreeks begrijpen en verwerken. Omgekeerd geldt dat machinetaal voor ons nauwelijks is te volgen en zeer omslachtig is in het gebruik.

DEBUG starten en beëindigen

Om batchbestanden in te voeren maakten we gebruik van COPY CON of EDIT; voor basic-programma's van de QBASIC-interpreter. Ook voor het invoeren van machinetaal gebruiken we een hulpmiddel.

Dat hulpmiddel is het programma DEBUG. De naam duidt erop dat het programma is ontwikkeld om fouten (in het Engels *bugs*) uit programma's te halen. U kunt het programma echter ook gebruiken om de inhoud van het geheugen van de PC te bekijken en te wijzigen, programma's stap voor stap uit te voeren en ten slotte om programma's te schrijven in machinetaal.

Als u DEBUG start vanaf de DOS-opdrachtregel, verschijnt een koppelstreepje als prompt op uw scherm. U zou nu bijvoorbeeld de opdracht kunnen invoeren die u een programma in machinetaal laat schrijven.

Wilt u DEBUG weer beëindigen, dan typt u een Q (van *QUIT*). De DOS-prompt verschijnt weer op uw scherm.

Machinetaal invoeren

Om u het werken in DEBUG zo makkelijk mogelijk te maken hebben we de volgende methode gekozen.

U voert alle assembler-opdrachten op de gebruikelijke wijze in in een tekstbestand, dus met behulp van COPY CON, EDIT of uw tekstverwerker. U werkt daarbij op vertrouwd terrein en kunt makkelijk wijzigingen aanbrengen. Daarna slaat u het be-

stand op de gebruikelijke manier op.

De tweede stap is het starten van DEBUG met het zojuist aangemaakte tekstbestand als invoer. Daarvoor maakt u gebruik van het omleidingssymbool '<'. De laatste opdracht in het tekstbestand is altijd een 'Q', zodat DEBUG ook weer wordt beëindigd. Het resultaat is een programmaatje dat u rechtstreeks kunt uitvoeren.

Bijzonderheden bij invoer

Zoals gewoonlijk is het absoluut noodzakelijk dat u de listing in het boek zonder fouten overneemt in het tekstbestand. Bij het invoeren van machinetaal geldt dat ook voor witregels tussen de opdrachten. Mocht DEBUG toch onverhoopt een fout ontdekken, dan verschijnt er een foutmelding op uw scherm (ook al haalt DEBUG de invoer uit uw tekstbestand). Breek in dat geval DEBUG af met Ctrl-Break en vergelijk de inhoud van uw tekstbestand met de listing in het boek.

Voorbeeld: een geluidssignaal

We maken een klein programma in assembler dat niet meer doet dan een geluidssignaal uitvoeren. Roep eerst de editor op om het programma BEEP.DEB te schrijven:

```
EDIT BEEP.DEB
```

Voer de volgende regels in, inclusief de witregel vóór de opdracht RCX.

```
A
MOV AH,2
MOV DL,7
INT 21
MOV AH,4C
INT 21

RCX
0B
nBEEP.COM
W
Q
```

Roep voor het vertalen van tekstbestand naar machinetaal DEBUG op de volgende manier op:

```
DEBUG <BEEP.DEB
```

Als DEBUG klaar is, beschikt u in de actuele directory over het programma BEEP.COM. Roep het programma op met de opdracht:

```
BEEP
```

Het systeem produceert een geluidssignaal.

2 Tips voor batchbestanden

2.1 Eenvoudig batchbestanden wijzigen

Het is nogal omslachtig om, steeds als u een batchbestand wilt wijzigen, eerst EDIT op te roepen en dan nog eens het batchbestand te laden. U kunt beide opdrachten combineren.

Uitwerking

Als u zich op het niveau van DOS bevindt, bestaat het probleem in feite niet: u activeert de directory met het batchbestand en roept EDIT op met de naam van het batchbestand als parameter. Het bestand wordt dan meteen geladen bij het starten van EDIT. Het bestand TEST.BAT roept u bijvoorbeeld als volgt op:

```
EDIT TEST.BAT
```

Het is belangrijk dat u de volledige naam en de extensie van het batchbestand opgeeft.

Als u de DOS Shell gebruikt, de grafische gebruikersinterface, kunt u batchbestanden ook op andere manieren in de editor laden. Daartoe koppelt u bestanden met de extensie .BAT met het bestand waarmee u EDIT start: EDIT.COM.

Activeer eerst de DOS Shell en de directory met de DOS-bestanden. Markeer in deze directory het bestand EDIT.COM. Open het menu File en kies de optie Associate. Er wordt een dialoogvenster op uw scherm geopend. In de bovenrand daarvan staat de EDIT.COM. Voer in het invoerveld voor de extensie '.BAT' in en druk op Enter. Alle bestanden met de extensie .BAT zijn nu aan EDIT.COM gekoppeld.

Als u nu in de DOS Shell een batchbestand kiest (dubbelklikken met de muis of markeren en op Enter drukken), wordt EDIT.COM geactiveerd en onmiddellijk het door u gekozen bestand geladen.

Voordat u de batchbestanden aan EDIT.COM koppelde, activeerde u een batchbestand zèlf door te dubbelklikken of op Enter te drukken. U kunt het batchbestand na de koppeling alleen nog starten vanuit de Shell door het menu File te openen en de optie Run te kiezen.

U kunt de koppeling weer ongedaan maken op dezelfde manier: markeer het bestand EDIT.COM en kies File en Associate. In het invoerveld van het dialoogvenster wist u nu echter de extensie .BAT. Daarna drukt u op Enter.

2.2 Parameters opsporen

Voor veel batchbestanden is het van belang te kunnen controleren of de gebruiker de nodige parameters heeft ingevoerd, zodat het batchbestand adequaat kan reageren.

Uitwerking

Voeg aan uw batchbestand de volgende regel toe om te reageren op ontbrekende parameters:

```
IF "%1" == "" GOTO GeenPara
```

Het stukje programma dat wordt gemarkeerd door de vlag 'GeenPara' ziet er als volgt uit:

```
:GeenPara
ECHO U dient bij deze opdracht een parameter op te geven
GOTO Einde
```

Vanzelfsprekend dient u ook de label 'Einde' aan uw programma toe te voegen. Schrijft u een batchbestand dat meer dan één parameter gebruikt, dan herhaalt u de sprongopdracht voor al die parameters (%2, %3, enzovoort).

Toelichting

Gebruikt u plaatshouders zoals %1, dan gebruikt u normaal gesproken geen aanhalingstekens. Om te kunnen controleren of er een parameter ontbreekt en dus de bijbehorende variabele in het batchbestand leeg blijft, zou de opdracht dus moeten luiden:

```
IF %1 ==   GOTO Fout
```

MS-DOS meldt dan echter een Syntax error, omdat u achter het gelijkteken niets ter vergelijking hebt opgegeven. Door nu aan beide zijden van het gelijkteken aanha-

lingstekens te gebruiken, ontstaat er een geldige vergelijking. Zou de parameter TEKST.TXT zijn, dan was de vergelijking:

```
IF "TEKST.TXT" == "" GOTO GeenPara
```

Aan de voorwaarde is niet voldaan, dus de sprongopdracht wordt niet uitgevoerd. Alleen bij een ontbrekende parameter gebeurt dat:

```
IF "" == "" GOTO GeenPara
```

2.3 Systeemvariabelen gebruiken in batchbestanden

MS-DOS is flexibel. Het besturingssysteem kan op allerlei manieren aan de omstandigheden worden aangepast. Belangrijke hulpmiddelen daarbij zijn de systeemvariabelen. U kunt die wijzigen met behulp van batchbestanden.

Uitwerking

U kunt te allen tijde een overzicht van de actuele systeemvariabelen opvragen met de opdracht SET. MS-DOS 5.0 gebruikt de volgende systeemvariabelen:

PATH — Het zoekpad voor programma's. MS-DOS zoekt programma's eerst in de actuele directory en daarna in de directories die zijn opgegeven in het zoekpad. Voor het probleemloos functioneren van een systeem dient eigenlijk altijd een pad naar de directory \DOS te bestaan.

APPEND — Het zoekpad voor gegevensbestanden, dus bestanden die door programma's met de extensie .EXE of .COM worden verwerkt.

PROMPT — Het teken dat u meldt dat MS-DOS gereed is voor invoer.

COMSPEC — Zoekpad en naam van het besturingssysteem. MS-DOS heeft deze gegevens nodig om bijvoorbeeld de besturing van applicaties over te kunnen nemen.

TEMP — Een directory die MS-DOS gebruikt voor het opslaan van tijdelijke bestanden en variabelen.

DIRCMD Een DOS-opdracht waarmee u de opdracht DIR aan uw wensen aanpast. Voert u de opdracht DIR in, dan voert MS-DOS die uit met inachtneming van de gegevens van DIRMCD. Zo zorgt de volgende opdracht ervoor dat het overzicht van een directory altijd per pagina op uw scherm verschijnt:

```
Set DIRMCD=/P
```

Deze variabelen worden ook wel *environment*-variabelen genoemd, omdat ze de omgeving definiëren waarin DOS werkt. Vanuit batchbestanden kunt u ze eenvoudig bereiken en wijzigen. Het onderstaande batchbestand COMSP.BAT controleert of er een besturingssysteem is opgegeven met de variabele COMSPEC. Zo niet, dan gebeurt dat alsnog.

```
@REM ****************************************************************
@REM * Batchbestand voor het definiëren van het besturingssysteem *
@REM ****************************************************************
@ECHO OFF
IF NOT "%COMSPEC%" == "" GOTO Verder
SET COMSPEC=C:\COMMAND.COM
ECHO Besturingssysteem: C:\COMMAND.COM
:Verder
ECHO ON
```

Toelichting

U kunt een systeemvariabele alleen in een batchbestand gebruiken als u de naam tussen aanhalingstekens zet. Om vast te stellen of de variabele is gezet, vergelijkt u '%COMSPEC%' met een lege string "". Als de variabele is gezet, springt het systeem naar het einde van het batchbestand. In het andere geval initialiseert het systeem de variabele met de verstekwaarde.

2.4 Eigen variabelen gebruiken in batchbestanden

Batchbestanden kunnen hun werk vaak pas goed doen als u meer mogelijkheden tot uw beschikking hebt dan de ingevoerde parameters en de systeemvariabelen u bieden; ze hebben dan eigen variabelen nodig om bijvoorbeeld tijdelijk gegevens in op te slaan.

Uitwerking

Met de DOS-opdracht SET definieert u moeiteloos nieuwe systeemvariabelen, die u later weer even makkelijk kunt verwijderen.

In het voorbeeld wijzigen we de systeemprompt zodanig dat datum, tijd, zoekpad en DOS-versie worden weergegeven. Aansluitend verschijnt de gebruikelijke DOS-prompt weer.

```
@REM *********************************************************
@REM * Batchbestand voor het tijdelijk wijzigen van de prompt *
@REM *********************************************************
@SET OLDPROMPT=%PROMPT%
PROMPT datum:$l$d$g tijd:$l$t$g $_pad:$l$p$g versie:$v

@SET PROMPT=%OLDPROMPT%
@SET OLDPROMPT=
```

Tip: Als u het batchbestand aanmaakt en start, kan dat aanleiding geven tot de volgende foutmelding:

```
Out of environment space
```

U lost dit probleem op door aan uw CONFIG.SYS de volgende regel toe te voegen:

```
SHELL=C:\COMMAND.COM /P /E:512
```

Vervang zo nodig 'C:\COMMAND.COM' door het zoekpad en de naam van het besturingssysteem dat u gebruikt. Met deze opdracht vergroot u de geheugencapaciteit van de environment van de verstekwaarde 160 tot 512 bytes. Indien nodig kunt u de environment instellen op een nog hogere waarde, bijvoorbeeld 2048.

Let op: Wijzigingen die u aanbrengt in het bestand CONFIG.SYS hebben pas effect na een reset van het systeem. U zult de computer dus eerst opnieuw moet starten, bijvoorbeeld met Ctrl-Alt-Del. Met de volgende opdracht bereikt u hetzelfde doel:

```
COMMAND /P /E:512
```

Het besturingssysteem wordt dan opnieuw geïnstalleerd. U mag deze op-

dracht echter alleen gebruiken als u zich werkelijk op DOS-niveau bevindt. Als u tijdelijk een applicatie hebt verlaten, leidt de opdracht tot een warboel in het geheugen en kunt u niet meer naar de applicatie terugkeren. Voor het tijdelijk activeren van DOS gebruiken applicaties verschillende opdrachten: SHELL bij Wordperfect, Bibliotheek en Systeem bij WORD 5.0. In de DOS Shell gaat u tijdelijk naar DOS met Shift-F9, Command Prompt. Al die opdrachten hebben gemeen dat u weer terugkeert naar uw applicatie met de opdracht EXIT.

Toelichting

Met de opdracht 'SET OLDPROMPT=%PROMPT%' definieert u een nieuwe systeemvariabele onder de naam OLDPROMPT; tegelijkertijd initialiseert u de variabele met de waarde van PROMPT. Daarmee voorkomt u dat u de oude waarde van de prompt verliest.

In de volgende opdracht definieert u een nieuwe systeemprompt. Daarbij worden tekens die worden voorafgegaan door een dollarteken, niet uitgevoerd naar het scherm, maar vervangen door de onderstaande tekens.

$l	het kleiner-dan teken: <
$g	het groter-dan teken: >
$d	de actuele datum
$t	de actuele tijd
$_	regelomslag; er wordt dus geen teken uitgevoerd naar het scherm
$p	drive en zoekpad
$v	de actieve versie van MS-DOS

De lege regel in het batchbestand is van wezenlijk belang. Hij zorgt ervoor dat DOS de nieuwe systeemprompt een keer op het scherm zet. Bij alle andere opdrachten hebben we weergave van de opdracht onderdrukt met behulp van de apestaart.

Met de regel '@SET PROMPT=%OLDPROMPT%' schakelen we de oude systeemprompt weer in. De eerstvolgende keer dat DOS een prompt op het scherm zet, wordt daar al van gebruik gemaakt.

De laatste regel, die er wat onvolledig uitziet, verwijdert de tijdelijke systeemvariabele OLDPROMPT uit de environment, zodat geheugenruimte vrij komt.

2.5 Gegevens opslaan in tijdelijke bestanden

Een batchbestand kan niet altijd alle gegevens onmiddellijk verwerken. Daarom is het handig om vanuit een batchbestand een bestand te kunnen aanmaken waarin ze tijdelijk worden opgeslagen.

Uitwerking

U leidt de gegevens die u tijdelijk wilt opslaan om naar een bestand met het pipesymbool. Later kunt u de gegevens dan weer ophalen en verwerken. Als het om aanzienlijke hoeveelheden gegevens gaat, werkt het onderstaande programmaatje uitstekend.

```
@REM *****************************************************
@REM * Dit batchbestand versnelt de uitvoer van bestanden *
@REM *****************************************************
IF EXIST \CHKDSK.DAT GOTO Uitvoer
@CHKDSK > \CHKDSK.DAT /V
:Uitvoer
@COPY \CHKDSK.DAT CON
```

Als u dit batchbestand voor de eerste keer start, duurt het tamelijk lang voordat de uitvoer op uw scherm komt. Daarna echter verschijnt de uitvoer onmiddellijk op uw scherm, omdat de gegevens niet eerst meer worden aangemaakt; alleen het bestand CHKDSK.DAT wordt gelezen.

Toelichting

U gebruikt de opdracht CHKDSK om een diskette of een harde schijf te controleren op fouten. De optie /V zorgt ervoor dat alle gecontroleerde bestanden op het scherm worden weergegeven. Het is een eenvoudige manier om de inhoud van de complete harde schijf in één keer te bekijken, inclusief alle subdirectories.

Het controleren van de bestanden is echter een nogal tijdrovende zaak. Daarom wordt de uitvoer van CHKDSK omgeleid naar het bestand CHKDSK.DAT. Vervolgens wordt dat bestand op het scherm weergegeven. Roept u het batchbestand een

tweede maal op, dan bestaat het bestand CHKDSK.DAT al en springt het programma naar de label ':Uitvoer'.

Tip: Als u bestanden wist of wijzigt of nieuwe bestanden toevoegt, is de inhoud van CHKDSK.DAT niet meer in overeenstemming met de werkelijkheid. U dient dan CHKDSK.DAT te wissen, zodat het batchbestand een nieuw bestand aan kan maken.

2.6 Waarschuwingen uitvoeren via batchbestanden

Er zijn opdrachten die nogal wat onheil kunnen veroorzaken als ze per ongeluk worden uitgevoerd, bijvoorbeeld de opdracht om alle tijdelijke bestanden met de extensie .TMP te wissen. De gebruiker dient dan eerst een controlevraag te beantwoorden.

Uitwerking

Het onderbreken van een batchbestand is vrij eenvoudig: neem een ECHO-opdracht op die wijst op het gevaar, en de opdracht PAUSE zodat de gebruiker met een toetsaanslag kan reageren. Als de gebruiker op dat moment op Ctrl-C drukt, wordt het batchbestand afgebroken. Het onderstaande programma maakt hiervan gebruik bij het wissen van tijdelijke bestanden.

```
@REM *****************************************
@REM * Dit batchbestand wist alle tijdelijke *
@REM *   bestanden in de actuele directory   *
@REM *****************************************
@ECHO Alle bestanden met de extensie .TMP worden gewist.
@ECHO Afbreken met Ctrl-C. Doorgaan?
@PAUSE
@DEL *.TMP
```

Toelichting

De opdracht PAUSE zet de volgende melding op het scherm:

```
Press any key to continue
```

Drukt u op Ctrl-C, dan verschijnt de melding:

```
Terminate batch job (Y/N)?
```

Druk op 'Y' om het bestand af te breken. Als u wilt afbreken met Ctrl-C, is het belangrijk eerst de Ctrl-toets in te drukken en goed ingedrukt te houden, anders ontvangt het systeem alleen de toets 'C' en interpreteert het die als teken om door te gaan met het wissen van de bestanden.

2.7 Batchbestanden beschikbaar maken

We hopen dat u in dit boek een groot aantal batchbestanden zó bruikbaar vindt dat u ze voortdurend bij de hand wilt hebben. Hieronder bespreken we hoe u daarvoor zorgt.

Uitwerking

Wat u doet, hangt af van de PC waarmee u werkt:

1 Werkt u met een PC met één diskettestation, dan zal de systeemschijf hoogstwaarschijnlijk vol staan met DOS-opdrachten en -programma's. Wellicht helpt een tweede, aanvullende systeemschijf u op weg. U kunt dan de DOS-bestanden en batchbestanden zo goed mogelijk verdelen over beide schijven. Hebt u meer dan 512 Kb werkgeheugen, dan kunt u een RAM-disk aanleggen en daar batchbestanden naar toe kopiëren. Voeg de RAM-disk toe aan het zoekpad achter PATH, dan zijn ze te allen tijde bereikbaar.

2 Beschikt u over een PC met twee diskettestations, dan zult u de DOS-systeemschijf in A: gebruiken. In B: kunt u dan een diskette plaatsen met de batchbestanden. In dat geval neemt u drive B: op in uw zoekpad.

3 Het makkelijkste is een en ander natuurlijk te verwezenlijken, als u beschikt over een harde schijf. Maak daartoe een directory C:\BAT aan en kopieer alle batchbestanden naar die directory. Laat het bestand AUTOEXEC.BAT wel in de hoofddirectory staan, anders kan het systeem het bij het starten niet vinden. Wijzig de opdracht PATH in het bestand AUTOEXEC.BAT. Neem C:\BAT als eerste zoekpad op, zodat batchbestanden voorrang krijgen boven programma's met de extensie .COM en .EXE die zich in latere zoekpaden bevinden.

2.8 Gegevens opvragen in een batchbestand

Het is niet altijd mogelijk alle noodzakelijke gegevens voor het uitvoeren van de opdrachten mee te geven als parameters bij het starten van het batchbestand. Tijdens de uitvoering van het bestand moeten gegevens kunnen worden ingevoerd.

Uitwerking

Beschikt u over de Norton Utilities, dan is het probleem snel opgelost. Een van de programma's van dit pakket, BE.EXE (*Batch-enhancer*), is onder andere voor dit doel ontworpen. Achter de opdracht BE ASK kunt u een tekst plaatsen tussen aanhalingstekens, gevolgd door een komma en daarachter kunt u een reeks toetsen opgeven waarop de gebruiker mag drukken ten antwoord. De volgorde is daarbij wel belangrijk, omdat de gegevens later met de opdracht ERRORLEVEL worden verwerkt.

Kopieer het programma BE.EXE naar de directory C:\BAT, zodat alle batchbestanden er gebruik van kunnen maken.

Het onderstaande programmaatje dient alleen ter illustratie van het gebruik van ASK.

```
@REM ***********************************************
@REM * Dit batchbestand gebruikt BE.EXE van NORTON *
@REM ***********************************************
@ECHO Ik stel een vraag
@BE ASK "J/N",JN
@IF ERRORLEVEL 2 GOTO N
@IF ERRORLEVEL 1 GOTO J
@GOTO :Einde
:J
@ECHO Ja
@GOTO :Einde
:N
@ECHO Nee
@GOTO :Einde
:Einde
```

Tip: In oudere versies van de Norton Utilities is de opdracht ASK ondergebracht in een apart bestand ASK.EXE. Kopieer dan ASK.EXE naar C:\BAT en vervang de opdracht BE ASK door ASK.

Beschikt u niet over de Norton Utilities, dan bestaat er toch nog een relatief makkelijke methode om gegevens op te vragen. U gebruikt een kleine programmaatje in machinetaal, dat dezelfde functie heeft. Voor het maken van dat programma hoeft u niet te beschikken over speciale kennis. Lees eventueel paragraaf 1.6 nog een keer door.

Voer de volgende regels met behulp van een editor (bijvoorbeeld EDIT) in onder de naam VRAAG.DEB.

```
A
MOV AX,0C07
INT 21
MOV AH,4C
INT 21

RCX
9
nVRAAG.COM
W
Q
```

Tip: De witregel is onderdeel van het programma; zou u hem niet toevoegen, dan werkt het programma niet.

Sla het bestand op en start het DOS-programma DEBUG met de volgende opdracht:

```
DEBUG <VRAAG.DEB
```

Als DEBUG klaar is, hebt u een klein zelfstandig uitvoerbaar programma VRAAG.COM dat u als volgt kunt gebruiken:

```
@REM ***************************************
@REM * Dit batchbestand gebruikt VRAAG.COM *
@REM ***************************************
@ECHO OFF
:Invoer
ECHO j(a) of n(ee)?
VRAAG
IF ERRORLEVEL 110 GOTO Nee
IF ERRORLEVEL 106 GOTO Ja
GOTO Invoer
:Ja
```

```
IF ERRORLEVEL 107 GOTO Invoer
ECHO Ja
GOTO Einde
:Nee
IF ERRORLEVEL 111 GOTO Invoer
ECHO Nee
GOTO Einde
:Einde
@ECHO ON
```

In plaats van 'j' en 'n' kunt u ook andere invoer opvragen. Achter ERRORLEVEL geeft u dan de desbetreffende ASCII-code op. Het is belangrijk dat u de ERRORLEVELS opvraagt in afdalende volgorde.

Toelichting

Het programma in machinetaal dat u met behulp van DEBUG hebt gemaakt, wacht op een toetsaanslag. Dan levert het de ASCII-code van het bijbehorende teken als foutcode. In het batchbestand kan die foutcode worden geïnterpreteerd met de opdracht ERRORLEVEL. Aan de gestelde voorwaarde is voldaan, als ERRORLEVEL groter is dan of gelijk is aan de vergeleken waarde. Daarom moeten de ASCII-codes ook in afdalende volgorde worden gecontroleerd om de juiste sprongopdracht tot gevolg te hebben.

Aansluitend wordt steeds gecontroleerd of ERRORLEVEL misschien groter is dan de opgegeven ASCII-waarde. Bleef die controle achterwege, dan zou u in plaats van 'n' ook 'o' kunnen typen als antwoord, omdat dan nog steeds aan de voorwaarde voor de sprongopdracht is voldaan.

3 MS-DOS 5.0 installeren

In dit hoofdstuk gaan we wat dieper in op de overstap naar MS-DOS 5.0. We geven u een aantal tips en trucs die het u wat makkelijker maken en mogelijke problemen kunnen voorkomen.

3.1 Update van MS-DOS x.x naar MS-DOS 5.0

Overstappen op een nieuwe versie van DOS is niet altijd even gemakkelijk. Hieronder gaan we dieper in op de mogelijke problemen.

Uitwerking

Werkt u op dit moment met MS-DOS 2.11 of hoger, dan is er niets aan de hand: u kunt de overstap maken met behulp van de upgrade-versie van MS-DOS 5.0. Die bevat een installatieprogramma dat u snel, moeiteloos en veilig op weg helpt.

Voorwaarden voor een standaardinstallatie

In principe kan er niet veel fout gaan, omdat alle belangrijke gegevens worden opgeslagen in een directory OLD_DOS. Met behulp van het programma UNINSTALL kunt u dan altijd terugkeren naar uw oude versie. Er zijn echter een aantal uitzonderlijke situaties die voor problemen kunnen zorgen:

- Er moet minstens 2,8 Mb vrij zijn op uw harde schijf. Als dat niet het geval is, kunt u bij de start van SETUP met de optie /M een minimale installatie opgeven die ongeveer 512 Kb nodig heeft. Met de optie /F wijkt u uit naar een installatie op diskettes. In beide situaties worden in ieder geval de gecomprimeerde bestanden gedecomprimeerd.

- Als u tot nu toe met een DOS-versie werkt, die partities groter dan 32 Mb niet toestaat, hebt u uw harde schijf wellicht verdeeld in meer dan één drive. MS-DOS 5.0 kan wel werken met harde schijven met meer capaciteit. U kunt van het moment van de update gebruik maken om MS-DOS 5.0 uw harde schijf te laten reorganiseren, zodat alle gegevens worden ondergebracht in één partitie. Dat levert u wel wat extra werk op: u moet van tevoren een backup maken van alle bestanden op de harde schijf, want bij de herverdeling van partities gaan

alle gegevens verloren. Na de update zult u de backup weer terug moeten schrijven naar de harde schijf.

- Sommige programma's zijn gemaakt om onder een bepaalde versie van DOS te werken. In dat geval moet u de mogelijkheid open houden het systeem te starten met die versie van DOS. Daartoe kunt u het beste voor de update een systeemdiskette maken, die u in die speciale gevallen gebruikt.

- Werkt u binnen een netwerk en installeert u MS-DOS 5.0 op een station, dan zult u na de update niet onmiddellijk toegang tot het netwerk hebben. Dat komt omdat de programma's die het netwerk beheren, gebonden zijn aan bepaalde versies van MS-DOS. Bij NOVELL heten die programma's bijvoorbeeld NET3.COM of NET4.COM, bedoeld voor respectievelijk de versies 3.x en 4.x van MS-DOS. Met behulp van de opdracht SETVER kunt u na de update dit probleem oplossen. Hieronder gaan we nog dieper op die opdracht in. Een andere mogelijkheid biedt het gebruik van het bij MS-DOS 5.0 meegeleverde programma NET5.COM, dat u echter eerst met EXPAND bruikbaar moet maken.

- In principe werken MS-DOS 5.0 en WINDOWS 3.0 goed samen. Er zijn echter een paar voorwaarden. Op een PC met 386-processor dient het bestand WINA20.386 bijvoorbeeld in de hoofddirectory te staan om WINDOWS zonder problemen te kunnen starten.

MS-DOS 5.0 installeren

Als aan de hierboven beschreven voorwaarden is voldaan, laat MS-DOS 5.0 zich heel eenvoudig installeren. U hebt de installatiediskette (van de upgrade-versie) nodig en één of twee lege diskettes. We gaan er bij het onderstaande van uit dat u werkt op een stand-alone PC en dat u beschikt over een recente backup van uw harde schijf. Als dat laatste niet het geval is, raden we u beslist aan eerst een backup te maken. U gaat als volgt te werk:

1 activeer drive A:

2 start het programma SETUP

3 bevestig de melding omtrent de systeemconfiguratie (tenzij die fout is)

4 beantwoord de vraag naar het netwerk met nee

5 beantwoord de vraag over het maken van een backup met nee

6 start de installatieprocedure

SETUP leest vervolgens alle belangrijke systeemgegevens van de oude DOS-versie en vraagt u een lege diskette te plaatsen waarop deze gegevens worden opgeslagen (UNINSTALL). Volg daarna de aanwijzingen op het scherm voor het plaatsen van de overige diskettes.

3.2 Nieuw bij MS-DOS 5.0

Het is van belang om na de update op de hoogte te zijn van de verschillen met versie 4.0 van MS-DOS. Een hele serie tips en trucs die voor MS-DOS 4.0 nog wel golden, zijn bij versie 5.0 overbodig geworden. Een aantal daarvan staat zelfs probleemloos werken in de weg.

Uitwerking

Hieronder volgt een opsomming van wijzigingen in MS-DOS 5.0 ten opzicht van versie 4.0:

- High memory is beschikbaar voor computers met een 80286-processor en minstens 64 Kb extended memory. U kunt de kernel, het BIOS en het residente gedeelte van COMMAND.COM opslaan in het HMA (High Memory Area), evenals een deel van het bestand HIMEM.SYS en de DOS-buffers.
- In het expanded memory kunt (en hoeft) u geen buffers meer aan te leggen.
- U kunt partities groter dan 32 Mb beheren zonder SHARE.EXE.
- Als u het besturingssysteem COMMAND.COM een tweede keer start, hebt u daarvoor maar weinig geheugenruimte nodig, omdat het grootste deel van de al aanwezige programmacode opnieuw kan worden gebruikt.
- MS-DOS 5.0 ondersteunt het beheer van Upper Memory Blocks (UMB's), dat wil zeggen geheugenfragmenten in het gebied tussen 640 Kb en één Mb. Daar kunt u device drivers en programma's onderbrengen met behulp van de opdrachten DEVICEHIGH en LOADHIGH. Op die manier spaart u ruimte uit in het werkgeheugen.
- DOS is vanaf versie 5.0 uitgerust met een aantal help-functies. Na de opdracht HELP verschijnt een overzicht van alle DOS-opdrachten op uw scherm. Voegt u de optie /? toe aan een opdracht, dan krijgt u informatie over de syntaxis en de functie van de opdracht.
- Het aantal functies bij de opdracht DIR is aanzienlijk uitgebreid. U kunt de be-

standen nu op allerlei manieren gesorteerd weergeven in allerlei formaten. Daarbij kan de opdracht ook subdirectories tonen en selecteren op grond van bestandsattributen.

- Met de opdracht ATTRIB kunt u nu ook de attributen SYSTEM en HIDDEN wijzigen.
- De opdracht FORMAT is zodanig aangepast, dat de gevolgen weer ongedaan kunnen worden gemaakt. Daarnaast kunt u kiezen voor een snelle formatteringsprocedure waarbij de te formatteren schijf niet eerst wordt gecontroleerd.
- Twee nieuwe opdrachten, UNDELETE en UNFORMAT, zorgen voor het herstellen van gewiste bestanden en het ongedaan maken van de FORMAT-procedure.
- Met behulp van het nieuwe hulpprogramma DOSKEY kunt u nu de opdrachten van de DOS-opdrachtregel opnieuw oproepen achter de prompt en daar wijzigen. Met DOSKEY kunt u ook een serie opdrachten bundelen tot een batchbestand en macro's aanmaken.
- U kunt het DOS-versienummer voor afzonderlijke programma's wijzigen. Daardoor kunt u ook programma's gebruiken die een specifiek versienummer vereisen.
- MS-DOS 5.0 ondersteunt een directory voor tijdelijke bestanden (bijvoorbeeld bestanden die worden aangemaakt in verband met het pipe-symbool of de DOS Shell). De directory kunt u vastleggen met de omgevingsvariabele TEMP.
- De nieuwe DOS Shell is aanzienlijk uitgebreid en gestroomlijnd. Een belangrijke uitbreiding is *Task swapper*, waarmee een eenvoudige vorm van multi-tasking mogelijk wordt. De tijdelijk niet-actieve programma's worden in hun actuele toestand opgeslagen op de harde schijf.
- In plaats van het vertrouwde GWBASIC wordt bij MS-DOS 5.0 nu QBASIC meegeleverd, een interpreter-versie van QUICKBASIC, die gebruikersvriendelijker is en beter presteert dan zijn voorganger.
- MS-DOS 5.0 ondersteunt de nieuwe 3,5 inch-diskettes met een capaciteit van 2,88 Mb.
- Er worden toetsenbord-drivers meegeleverd voor Oosteuropese landen.

Wellicht vindt u dit allemaal een beetje veel. Daarom zullen we de belangrijkste wijzigingen wat uitvoeriger bespreken.

3.3 Beginnen met MS-DOS 5.0

Met de installatie van MS-DOS 5.0 bent u er nog niet. We geven u een aantal tips voor het configureren van het systeem op uw computer waardoor u uw nieuwe besturingssysteem optimaal kunt gebruiken.

Uitwerking

Hieronder volgen enkele interessante en effectieve wijzigingen in MS-DOS. Om te beginnen maakt MS-DOS 5.0 niet alleen beter gebruik van het werkgeheugen, het kan ook veel beter overweg met geheugenuitbreidingen. We willen de mogelijkheden laten zien aan de hand van twee veel gebruikte configuraties.

Geheugenbeheer voor een PC met een 80286-processor en minstens 1 Mb geheugen

Voeg de beide volgende opdrachten toe aan uw bestand CONFIG.SYS:

```
DEVICE=HIMEM.SYS
DOS=HIGH
```

De opdrachten zorgen ervoor dat DOS het geheugen boven 640 Kb (normaal gesproken extended memory) beheert als XMS-geheugen en zelf de eerste 64 Kb daarvan gebruikt. Dat levert u een uitbreiding van het werkgeheugen op van ongeveer 50 Kb.

Geheugenbeheer voor een PC met een 80386-processor en minstens 1 Mb geheugen

MS-DOS kan het geheugen van een PC met een 80386-processor nog effectiever gebruiken. In het gebied tussen 640 Kb en 1 Mb bevinden zich altijd stukken geheugenruimte die niet worden gebruikt. MS-DOS 5.0 kan deze Upper Memory Blocks adresseren en gebruiken als u de volgende opdrachten toevoegt aan uw bestand CONFIG.SYS:

```
DEVICE=HIMEM.SYS
DEVICE=EMM386.EXE
DOS=HIGH
```

Tip: Bij deze instelling wordt het geheugen boven de grens van 1 Mb automatisch gedefinieerd als expanded memory. Wilt u met WINDOWS 3.0 in de standaardmodus of de extended modus voor 386 werken, dan dient u aan de opdracht 'DEVICE=EMM386.EXE' de optie NOEMS toe te voegen, omdat WINDOWS in de standaardmodus of de extended modus alleen extended memory kan adresseren:

```
DEVICE=EMM386.EXE NOEMS
```

Los van dit probleem moeten programma's zoveel mogelijk worden geladen in het UMB-gebied. Voor device drivers gebruikt u daartoe in plaats van de opdracht DEVICE de nieuwe opdracht DEVICEHIGH; voor programma's de opdracht LOADHIGH. De muis-driver MOUSE.SYS laadt u bijvoorbeeld in het UMB-gebied door de volgende opdracht aan uw bestand CONFIG.SYS toe te voegen:

```
DEVICEHIGH=MOUSE.SYS
```

Het programma MOUSE.COM laadt u in dit gebied door de volgende opdracht toe te voegen aan uw bestand AUTOEXEC.BAT:

```
LOADHIGH MOUSE.COM
```

Tip: Wellicht verbaast het u dat de opdracht DEVICEHIGH uit meer dan de toegestane acht tekens bestaat. Het gaat echter om een interne opdracht die niet als programmabestand is opgeslagen op schijf. Alleen dan doet zich namelijk een probleem voor, omdat DOS geen bestandsnamen accepteert die uit meer dan acht tekens bestaan.

Gegevensopslag en herstel van gewiste gegevens

In MS-DOS 5.0 is het mogelijk geworden om van belangrijke systeemgebieden (hoofddirectory en FAT) een reservekopie te maken. Daartoe gebruikt u de voor versie 5.0 nieuwe opdracht MIRROR. Wilt u onmiddellijk na het starten van het systeem de systeemgebieden van drive C: opslaan, dan voegt u de volgende opdracht toe aan uw bestand AUTOEXEC.BAT:

```
MIRROR C:
```

U kunt de werking van het programma nog uitbreiden met een belangrijke functie die gegevens opslaat van bestanden als u die wist. Het herstellen van een gewist bestand wordt daardoor veel makkelijker. Voegt u aan de bovenstaande opdracht de opties /tc toe, dan wordt een reservekopie van de systeemgebieden van drive C: gemaakt en worden de gegevens van gewiste bestanden bijgehouden in een protocolbestand.

```
MIRROR C: /tc
```

In noodgevallen kunt u de systeemgebieden herstellen met UNFORMAT; gewiste bestanden herstelt u met UNDELETE.

De nieuwe DOS Shell

Beschikte u tot nu toe over MS-DOS 3.3 en hebt u zo langzamerhand genoeg van de DOS-opdrachtregel, dan kunt u nu de nieuwe DOS Shell gebruiken. Bent u al vertrouwd met de DOS Shell van MS-DOS 4.0, dan zult u even moeten wennen. U zult echter merken dat de nieuwe shell logischer in elkaar zit, meer overeenkomsten vertoont met vergelijkbare programma's en makkelijker te hanteren is.

Een bijzonder interessante functie van de nieuwe shell is de *Task swapper* die het mogelijk maakt een programma te onderbreken en een ander programma te activeren. In principe activeert u de functie door in het menu Options de opdracht Task swapper te kiezen.

Tip: Zie het hoofdstuk over de DOS Shell voor meer informatie over het gebruik van de Task swapper.

Opdrachten herhalen

In oudere versies van DOS kon u alleen opdrachten achter de systeemprompt herhalen en wijzigen met behulp van de functietoetsen, meestal een omslachtig en ondoorzichtig proces. MS-DOS 5.0 beschikt over het hulpprogramma DOSKEY, dat die taak een stuk makkelijker maakt. Voeg de volgende regel toe aan het bestand AUTOEXEC.BAT:

```
DOSKEY
```

Die moet eventueel worden voorafgegaan door drive en zoekpad. Aansluitend beschikt u ook over de cursortoetsen om de opdrachten achter de systeemprompt te bewerken. Met de toetsen cursor naar links en cursor naar rechts verplaatst u de cursor nu zonder dat tekens worden gewist. Met de toetsen cursor omhoog en cursor omlaag haalt u andere opdrachten op uit de buffer van DOSKEY. DOSKEY heeft nog veel meer te bieden. Zie daartoe de paragraaf in dit boek die aan de opdracht is gewijd.

Help-functies van DOS

In MS-DOS 5.0 zijn ook op het niveau van het besturingssysteem help-functies toegevoegd. Wilt een vrij algemeen overzicht van alle opdrachten die MS-DOS 5.0 u biedt, dan typt u de opdracht HELP. Meer gedetailleerde help-schermen roept u op met HELP gevolgd door de naam van de opdracht of door aan de opdracht de optie /? toe te voegen. Wilt u bijvoorbeeld meer weten over de opdracht FORMAT, dan typt u:

```
HELP FORMAT
```

of

```
FORMAT /?
```

3.4 Problemen vermijden met versienummers

Programma's kunnen niet altijd zonder problemen draaien onder alle versies van MS-DOS. Sommige programma's vragen daarom het versienummer op en functioneren niet als dat niet klopt. Helaas leidt dat er ook toe dat na iedere update sommige programma's helemaal niet meer willen draaien. Met de opdracht SETVER valt daar wat aan te doen.

Uitwerking

Als een programma niet draait onder MS-DOS 5.0, omdat het bijvoorbeeld een ander soort geheugenbeheer vereist, zult u of een update van het programma moeten aanschaffen, of een systeemdiskette moeten aanmaken van de oude DOS-versie. Wilt u dan het bewuste programma gebruiken, dan start u het systeem opnieuw vanaf de diskette.

Als het echter alleen om het versienummer gaat en het programma bijvoorbeeld alleen wil werken als 4.0 wordt opgegeven als versienummer, is er weinig aan de hand. Met de opdracht SETVER tovert u de applicatie elk gewenst versienummer voor. Het besturingssysteem verandert in het geheel niet; er hangt alleen een ander bordje op de deur.

Veronderstel dat u werkt op een netwerkstation, dat werkt met het programma NET4.COM. Dat programma is voor een netwerk even belangrijk als COMMAND.COM voor een stand-alone computer. NET4.COM werkt alleen onder MS-

DOS 4.0. We gaan ervan uit dat u het systeem start vanaf de harde schijf. U dient dan het volgende te doen om onder MS-DOS 5.0 met NET4.COM te kunnen werken:

1 Voeg de volgende regel toe aan het bestand CONFIG.SYS:

```
DEVICE=SETVER.EXE
```

Voeg zo nodig een zoekpad toe.

2 Roep SETVER een keer op vanaf de systeemprompt met de programmanaam en het versienummer als parameters:

```
SETVER NET4.COM 4.00
```

Er volgt een melding die u wijst op de gevaren van het gebruik van programma's onder een ander versienummer, maar de versietabel wordt wel aangepast. MS-DOS bewaart de versietabel op de harde schijf in de systeembestanden. Omdat die pas bij een herstart opnieuw worden geladen, wordt het gewijzigde versienummer ook dan pas actief.

Tip: Wilt u de aanpassing van het versienummer weer ongedaan maken, dan voert u de opdracht nogmaals in, nu met de parameter /D (van *Delete*):

```
SETVER NET4.COM /D
```

4 DOS in uw eigen huisstijl

Ieder programma dat u installeert, kent een groot aantal functies dat u zelden gebruikt en dus snel vergeet. Het zal met MS-DOS 5.0 niet veel anders gaan. Tegen de tijd dat u zo'n functie nodig hebt, besteedt u veel tijd aan het zoeken in handboeken. We willen u in dit hoofdstuk laten zien hoe u belangrijke informatie op elk gewenst moment op uw scherm kunt laten weergeven. Daarnaast bespreken we weer een aantal tips waarmee u MS-DOS kunt aanpassen aan uw eigen omstandigheden. Veel van die tips maken gebruik van batchbestanden.

4.1 Een eigen handleiding samenstellen

Vooral bij opdrachten die u weinig gebruikt, komt het vaak voor dat u de syntaxis niet meer precies weet of dat u vergeten bent hoe u bepaalde opties kunt opgeven. MS-DOS helpt u snel verder met de geïntegreerde help-schermen. Voor eigen programma's kunt u zelf help-schermen maken.

Uitwerking

In principe is in 5.0 aan alle opdrachten een help-functie gekoppeld. Roept u de opdracht op met de optie /?, dan biedt MS-DOS u het bijbehorende help-scherm. Om even te zien welke mogelijkheden u hebt bij de opdracht FORMAT, geeft u dus de opdracht:

```
FORMAT /?
```

MS-DOS 5.0 biedt u ook een nieuwe opdracht HELP die, gecombineerd met de naam van de opdracht, hetzelfde effect heeft:

```
HELP FORMAT
```

Weet u niet meer precies welke opdracht u nodig hebt voor een bepaalde handeling, dan kunt u de opdracht HELP ook zonder begeleidende opdrachtnaam invoeren. MS-DOS geeft u dan een overzicht van alle DOS-opdrachten met een omschrijving in één regel.

Help-functies wijzigen en uitbreiden

De informatie die MS-DOS u biedt, is vanzelfsprekend erg algemeen van aard. Niets weerhoudt u er echter van eigen help-teksten te maken. U dient dan aan de volgende drie voorwaarden te voldoen:

1 Maak een aparte directory aan voor batchbestanden, bijvoorbeeld C:\BAT.

2 Voeg aan de opdracht PATH in het bestand AUTOEXEC.BAT een zoekpad toe naar deze directory. Zorg er daarbij voor dat dit zoekpad vóór het zoekpad naar de directory met DOS-bestanden staat, dus:

```
PATH C:\BAT;C:\DOS
```

3 Maak voor iedere opdracht die u wilt voorzien van een eigen help-scherm een apart batchbestand.

Voor de opdracht FORMAT zou u het volgende help-bestand kunnen aanmaken onder de naam FORMAT.BAT:

```
@REM ***************************
@REM * Help-scherm voor FORMAT *
@REM ***************************
@ECHO OFF
IF "%1" == "/?" GOTO Help
FORMAT.COM %1 %2 %3 %4 %5 %6 %7 %8 %9
GOTO Einde
:Help
ECHO *****************************************************
ECHO * Formatteert een schijf voor gebruik onder MS-DOS
ECHO.*
ECHO * FORMAT drive: [/V[:label]] [/Q] [/U] [F:cap] [/B | /S]
ECHO * FORMAT drive: [/V[:label]] [/Q] [/U] [T:sporen /N:sectoren]
                                                        [/B | /S]
ECHO * FORMAT drive: [/V[:label]] [/Q] [/U] [/1] [/4] [/B | /S]
ECHO * FORMAT drive: [/Q] [/U] [/1] [/4] [/8] [/B | /S]
ECHO.*
ECHO * /V[:label]  naam (label) voor de diskette
ECHO * /Q          formatteert met Quickformat
ECHO * /U          formatteert definitief
ECHO * /F:cap      capaciteit van de te formatteren diskette;
                                                          geldige
```

```
ECHO *                waarden: 160, 180, 320, 360, 720, 1.2, 1.44, 2.88
ECHO * /B             reserveert ruimte voor systeembestanden
ECHO * /S             kopieert systeembestanden naar de diskette
ECHO * /T:sporen      formatteert met het opgegeven aantal sporen
ECHO * /N:sectoren formatteert met het opgegeven aantal sectoren
ECHO * /1             formatteert alleen kant 1 van de diskette
ECHO * /4             formatteert een diskette met 360 Kb in een drive
ECHO *                met een hogere capaciteit
ECHO * /8             formatteert met acht sectoren per spoor
ECHO *
ECHO * formatteer 360 Kb-diskettes met FORMAT /4 of FORMAT /F:360
ECHO *
ECHO ********************************************************
:Einde
@ECHO ON
```

U roept het bestand op met de gebruikelijke opdracht, dus bijvoorbeeld:

```
FORMAT A:
```

Wilt u daarbij hulp, dan voegt u de optie /? toe:

```
FORMAT /?
```

Het resultaat ziet u in de afbeelding op de volgende pagina.

Toelichting

Als u een opdracht invoert, zoekt DOS in alle zoekpaden die u met de opdracht PATH hebt opgegeven, naar die opdracht. De zoekpaden worden daarbij netjes in volgorde afgewerkt. Als C:\BAT het eerste zoekpad is, vindt DOS daar al de gezochte opdracht, ook al is dat een batchbestand. Het systeem zoekt dus niet verder naar FORMAT.COM.

In het batchbestand zelf moet bij de opdracht om het eigenlijke programma op te roepen wel de extensie .COM zijn toegevoegd, anders zou het zoeken opnieuw uitkomen bij het batchbestand FORMAT.BAT.

```
FORMAT drive: [/V[:label]] [/Q] [/U] [F:cap] [/B | /S]
FORMAT drive: [/V[:label]] [/Q] [/U] [T:sporen /N:sectoren] [/B | /S]
FORMAT drive: [/V[:label]] [/Q] [/U] [/1] [/4] [/B | /S]
FORMAT drive: [/Q] [/U] [/1] [/4] [/8] [/B | /S]

/V[:label]   naam (label) voor de diskette
/Q           formatteert met Quickformat
/U           formatteert definitief
/F:cap       capaciteit van de te formatteren diskette: geldige
             waarden: 160, 180, 320, 360, 720, 1.2, 1.44, 2.88
/B           reserveert ruimte voor systeembestanden
/S           kopieert systeembestanden naar de diskette
/T:sporen    formatteert met het opgegeven aantal sporen
/N:sectoren  formatteert met het opgegeven aantal sectoren
/1           formatteert alleen kant 1 van de diskette
/4           formatteert een diskette met 360 Kb in een drive
             met een hogere capaciteit
/8           formatteert met acht sectoren per spoor

formatteer 360 Kb-diskettes met FORMAT /4 of FORMAT /F:360



D:\DOSTT>
```

Help-scherm voor de opdracht FORMAT

Tip: Als C:\DOS de actuele directory is, werkt het oproepen van het batchbestand niet. DOS zoekt altijd eerst in de actuele directory naar de opdracht en daarna pas in de aangegeven zoekpaden. Wilt u ook in de directory \DOS van de help-schermen gebruik kunnen maken, dan zult u daar de bestanden moeten hernoemen, bijvoorbeeld door ze vooraf te laten gaan door een streepje: '_'. In het batchbestand wijzigt u dan de oproep voor het eigenlijke programma als volgt:

```
C:\DOS\_FORMAT %1 %2 %3 %4 %5 %6 %7 %8 %9
```

In MS-DOS 5.0 kunt u echter ook met behulp van DOSKEY een macro aanmaken, die zelfs nog prioriteit heeft boven interne DOS-opdrachten en dus in ieder geval voor het (externe) FORMAT.COM wordt verwerkt. Om er zeker van te zijn dat de opdracht FORMAT altijd het batchbestand oproept, kunt u de volgende macro aanmaken:

```
DOSKEY FORMAT=C:\BAT\FORMAT.BAT %1 %2 %3 %4 %5
```

Het is slechts een van de gebruiksmogelijkheden van DOSKEY. We gaan dieper op de opdracht in in het hoofdstuk dat gewijd is aan het invoeren van opdrachten.

Teksten maken voor help-schermen

Vanzelfsprekend zult u de teksten eerst moeten schrijven. Daarbij kunt u natuurlijk gebruik maken van de teksten die DOS u al biedt. Het is verstandig die help-teksten eerst om te leiden naar een bestand:

```
FORMAT /? >FORMAT.HLP
```

Vervolgens staat uw help-tekst in het bestand FORMAT.HLP.

De structuur van DOSHELP.HLP

De nieuwe DOS-opdracht HELP komt op twee manieren aan zijn gegevens:

- In het bestand DOSHELP.HLP is een overzicht opgeslagen van alle DOS-opdrachten, voorzien van een omschrijving in een of twee regels. U kunt dit bestand met de opdracht TYPE op uw scherm weer laten geven. Vraagt u het bestand op met de opdracht HELP, dan wordt het per pagina op uw scherm weergegeven.
- Roept u HELP op met een opdracht als parameter en maakt de opdracht deel uit van het overzicht in DOSHELP.HLP, dan roept HELP zelf die opdracht op met optie /?. HELP geeft de vraag om help door aan de opdracht. Als de opdracht geen deel uitmaakt van het bestand, meldt HELP dat er voor de opdracht geen help-schermen ter beschikking staan.

DOSHELP.HLP is erg doorzichtig opgebouwd. Regels met commentaar worden voorafgegaan door een apestaart (@). De namen van opdrachten staan steeds aan het begin van de regel. De toelichting bij de opdracht staat daar dan achter, gescheiden door een spatie of tab. Als de toelichting meerdere regels beslaat, wordt ingesprongen.

Tip: De opdracht HELP werkt helaas alleen met bestanden met de extensie .COM of .EXE, niet met batchbestanden.

Eigen programma's toevoegen aan DOSHELP.HLP

Hebt u zelf een uitvoerbaar programma (.COM of .EXE) ontwikkeld, dan kunt u dit onderbrengen bij de help-functies van DOS. Daartoe moet uw programma de optie /? ondersteunen. De help-schermen van DOS zijn steeds volgens hetzelfde stramien opgebouwd. Eerst wordt de opdracht in een of twee regels kort omschreven. Na een regel wit volgt de beschrijving van de syntaxis en na nog een regel wit volgt de toelichting op de parameters en opties, eventueel nog gevolgd door overig commentaar.

Voeg aan het bestand DOSHELP.HLP een eigen regel toe. De opdrachten in dat bestanden zijn alfabetisch gesorteerd en u dient ervoor te zorgen dat uw opdracht op de juiste plaats komt te staan. De opdracht moet aan het begin van de regel staan; de toelichting volgt daarachter na een spatie.

4.2 Parameters ongeldig verklaren voor externe opdrachten

Sommige opdrachten richten groot onheil aan als ze in een moment van onachtzaamheid met de verkeerde parameters worden uitgevoerd. Te denken valt aan FORMAT C: of DEL *.*. Het zou mooi zijn als dergelijke opdrachten konden worden tegengehouden. Een controle van parameters biedt wat meer zekerheid.

Uitwerking

Als u op de hierboven beschreven manier help-bestanden hebt aangemaakt, wordt het erg eenvoudig om de parameters te controleren en in bepaalde gevallen het programma af te breken. We willen dat laten zien aan de hand van de opdracht FORMAT B:. We laten C: hierbij buiten beschouwing: een foutje zou te kostbaar kunnen worden. In het batchbestand uit de vorige paragraaf voegt u na de regel 'ECHO OFF' de volgende regels toe:

```
IF "%1" == "b:" GOTO Verboden
IF "%1" == "B:" GOTO Verboden
```

Na de laatste ECHO-regel voegt u de volgende regels toe:

```
GOTO Einde
:Verboden
ECHO Deze optie is niet toegestaan!
```

```
GOTO Einde
:Einde
@ECHO ON
```

Met deze regels verhindert u het formatteren van diskettes in drive B:. Zorg ervoor dat er geen diskette in de drive zit en probeer de opdracht uit:

```
FORMAT B:
```

Als u niets fout hebt gedaan, meldt het systeem:

```
Deze optie is niet toegestaan!
```

Het batchbestand werkt. U kunt nu 'B:' en 'b:' in het batchbestand wijzigen in 'C:' en 'c:'. Hebt u stalen zenuwen, dan kunt het batchbestand nogmaals uitproberen met de opdracht:

```
FORMAT C:
```

We raden u echter aan de opdracht verder met rust te laten. Wellicht komt er vanzelf een moment waarop u met een zucht van verlichting constateert dat u die ooit geschreven regels er niet voor niets aan toe hebt gevoegd.

Toelichting

Het is belangrijk om zowel 'B:' als 'b:' te controleren, omdat de opdracht IF onderscheid maakt tussen hoofdletters en kleine letters. Overigens is in MS-DOS 5.0 het formatteren van een harde schijf minder gevaarlijk dan vroeger. Toen was u echt al uw gegevens kwijt; nu kunt u met de opdracht UNFORMAT het formatteringsproces ongedaan maken.

4.3 Parameters ongeldig verklaren voor interne opdrachten

Omdat de veiligheidsklep in de vorige paragraaf is opgenomen in een batchbestand, werkt hij niet altijd. Bevindt u zich in de directory \DOS, dan wordt niet het batchbestand opgeroepen maar FORMAT.COM, met alle gevolgen vandien. Met het programma DOSKEY kunt u macro's maken die interne opdrachten opvangen en zodoende beveiligen.

Uitwerking

U kunt met het nieuwe programma DOSKEY niet alleen opdrachten redigeren achter de systeemprompt, maar ook macro's aanmaken. In principe is een macro een afkorting voor een opgeslagen serie opdrachten. Daarin onderscheidt hij zich dus niet zo zeer van een batchbestand.

Het grote verschil is dat een macro de hoogste prioriteit krijgt en zelfs nog vóór interne DOS-opdrachten wordt verwerkt. Nadat u DOSKEY start, is de rangorde als volgt:

1 DOSKEY-macro's

2 interne DOS-opdrachten (bijvoorbeeld DIR of DEL)

3 .COM-bestanden

4 .EXE-bestanden

5 batchbestanden

Tip: U kunt de rangorde doorkruisen door bij een opdracht ook de extensie te typen. Typt u bijvoorbeeld FORMAT.BAT, dan zal altijd het batchbestand worden opgeroepen en niet FORMAT.COM.

Om te verhinderen dat een gebruiker de opdracht DEL uitvoert, kunt u de volgende macro aanmaken:

```
DOSKEY DEL=ECHO Het wissen van bestanden is niet toegestaan.
```

Typt u vervolgens de opdracht achter de systeemprompt, dan wordt niet de interne opdracht uitgevoerd, maar verschijnt de ECHO-boodschap op het scherm. Daarbij is het wat storend dat DOS de melding twee keer op het scherm zet: één keer als opdracht (ongeveer analoog aan de opdracht ECHO in batchbestanden) en één keer als resultaat van de opdracht.

```
C:\>del
C:\>echo DEL Het wissen van bestanden is niet toegestaan
DEL Het wissen van bestanden is niet toegestaan

C:\>
```

Het wissen van bestanden verhinderd

Langs een omweggetje kunt u dat probleem oplossen: roep met de macro een batchbestand op dat de boodschap op het scherm zet. U gaat daarbij als volgt te werk:

1 Maak een batchbestand aan onder de naam DEL<Alt-255>.BAT. De naam zal u wat vreemd voorkomen. De eerste drie letters ('DEL') moet u letterlijk typen; daarna houdt u de Alt-toets ingedrukt, terwijl u op het numerieke gedeelte van het toetsenbord de toetsen '2', '5' en '5' typt. Samen vormen die het onzichtbare teken met de ASCII-code 255. De extensie ('.BAT') typt u weer letterlijk. Neem als inhoud van het batchbestand de volgende regels over:

```
@ECHO OFF
ECHO Het wissen van bestanden is niet toegestaan.
@ECHO ON
```

2 Maak een macro met de naam DEL, die als taak heeft het batchbestand te starten:

```
DOSKEY DEL=DEL<Alt-255>
```

Ook in dit geval gaat het weer om het teken met de ASCII-code 255.

De hinderlijke dubbele schermuitvoer is nu verdwenen. Wilt u elke vorm van schermuitvoer voorkomen, dan kunt op vergelijkbare wijze een batchbestand starten met de naam '<Alt-255>-<Alt-255>-<Alt-255> .BAT'. Een dergelijke onzichtbare bestandsnaam leidt natuurlijk wel gauw tot verwarring, vooral als u besluit voor meerdere batchbestanden het teken Alt-255 te gebruiken.

4.4 Een geluidssignaal uitvoeren via een batchbestand

Ter ondersteuning van een melding op het scherm en een controlevraag bij de uitvoering van gevaarlijke opdrachten kan het systeem nog eens extra aandacht vragen met een geluidssignaal.

Uitwerking

Het is erg eenvoudig om in MS-DOS een geluidssignaal te genereren. Achter de systeemprompt typt u:

```
Ctrl-G
```

Dat komt overeen met het teken met ASCII-code 7. Alle tekens met een ASCII-code kleiner dan 32 kunt u op die manier gebruiken: de letter G is de zevende letter van het alfabet.

Het volgende batchbestand illustreert het gebruik van het geluidssignaal:

```
ECHO OFF
ECHO Let op! De diskette in station A: wordt geformatteerd
ECHO Breek af met Ctrl-C. Doorgaan?
ECHO ^G
PAUSE
FORMAT A:
CHKDSK A:
ECHO ON
```

Tip: Voer dit batchbestand niet in met een tekstverwerker of met EDIT. Ctrl-G werkt dan namelijk niet. Gebruik in dit geval COPY CON.

Als u het bestand met TYPE op uw scherm weer laat geven, wordt de opdracht 'ECHO ^G' gewoon uitgevoerd: er klinkt een pieptoon. U vermijdt dat door die opdracht in z'n eentje in een klein batchbestandje te zetten onder de naam BEEP.BAT. In het eigenlijke batchbestand vervangt u de opdracht voor het geluidssignaal door de opdracht waarmee het sub-batchbestand oproept:

```
CALL BEEP.BAT
```

4.5 Geheugensteuntjes

De één vergeet de ASCII-code voor een e met trema, de ander weet steeds maar niet welke opties hij bij het formatteren moet opgeven. Veel gebruikers hangen hun PC met briefjes vol met dit soort meldingen. Een voor u op maat gemaakt help-scherm kan u van die briefjescultuur verlossen.

Uitwerking

Ook voor de oplossing van dit probleem maken we weer gebruik van batchbestanden. Het onderstaande voorbeeld vereist drie batchbestanden: HULP.BAT, HULP1.TXT en HULP2.TXT. Het illustreert tegelijkertijd hoe u de uitvoer per pagina naar het scherm kunt sturen. Als eerste het bestand HULP.BAT:

```
@REM ***************************************
@REM * Persoonlijk help-scherm voor MS-DOS *
@REM ***************************************
@ECHO OFF
CLS
COPY >NUL C:\BAT\HULP1.TXT CON
PAUSE
CLS
COPY >NUL C:\BAT\HULP2.TXT CON
PAUSE
CLS
```

Vervolgens de beide bestanden met de teksten. Aan de inhoud van het batchbestand hierboven kunt u zien dat we ervoor hebben gekozen die bestanden op te slaan in de directory C:\BAT. Beide tekstbestanden tellen inclusief de kaders 22 regels, zodat er op het scherm nog net ruimte overblijft voor DOS om de opdracht PAUSE uit te voeren.

```
                        Help voor MS-DOS
PIPE      Alt-124  |
Backslash Alt-92   \
Omleiden met < en > of >>


                        Belangrijkste batchbestanden
WAARBST.BAT   doorzoekt alle directories naar een bestand
DOALL.BAT     voert een opdracht uit voor àlle bestanden
```

DO.BAT	voert een opdracht uit binnen één directory
ASCII.BAT	plaatst een ASCII-tabel op het scherm
VRAAG.COM	maakt controlevragen mogelijk via ERRORLEVEL
HULP.BAT	geeft deze help-schermen weer
WAARDIR.BAT	zoekt een subdirectory
FINDALL.BAT	zoekt een bestand; uitvoer naar ALLBST.DAT
FINDMORE.BAT	selecteert bestanden op grond van criteria
WW.BAT	beveiligt ingeschakelde PC bij afwezigheid
VET.BAT	geeft tekst vet weer op het scherm
NORMAAL.BAT	geeft tekst normaal weer op het scherm
INVERT.BAT	geeft tekst invers weer op het scherm

Toelichting

Als u meer dan 22 regels in dit bestand onderbrengt, zal het scherm scrollen totdat alle tekst geplaatst is; de bovenste regels verdwijnen daardoor alweer van het scherm voordat u ze kunt lezen. Gebruikt u niet meer dan 22 regels, dan zorgt PAUSE ervoor dat het scherm pas wordt gewist als u op een toets drukt.

Het is bovendien van belang dat u de opdracht ECHO OFF aan het begin niet vergeet. Om de meldingen '1 file(s) copied' te onderdrukken, leiden we de uitvoer van die opdracht om naar het 'apparaat' NUL, die altijd tot gevolg heeft dat de uitvoer niet wordt weergegeven of opgeslagen. Het apparaat CON daarentegen bestaat wel; bij invoer van gegevens wordt het toetsenbord ermee aangeduid; bij uitvoer het scherm.

We gebruiken twee tekstbestanden in plaats van één, omdat bepaalde tekens ('<', '>' en '|') in batchbestanden verboden zijn. U kunt ze niet met ECHO uitvoeren naar het scherm. Omdat de tekens wel op het scherm moeten worden weergegeven, maken we ze onderdeel van een tekstbestand, waarin ze wel gewoon worden weergegeven. Zie ook paragraaf 1.4.

4.6 Help-schermen oproepen met F1

Veel professionele applicaties stoppen hun help-functie achter de functietoets F1. Ook binnen MS-DOS kunt u die lijn doortrekken met behulp van batchbestanden.

Uitwerking

In principe kunt u zelf aan alle toetsen tekens koppelen. Drukt u vervolgens op zo'n toets, dan plaatst MS-DOS die tekens op het scherm in plaats van het opschrift op de toets zelf.

Daartoe dient u de device driver ANSI.SYS op te nemen in het bestand CONFIG.SYS. Alleen met behulp van ANSI.SYS kunt u toetsen hernoemen.

We gaan ervan uit dat u de help-bestanden van de vorige paragraaf hebt aangelegd. Voeg dan nu de volgende drie regels toe aan het bestand AUTOEXEC.BAT om de help-schermen met F1 op te roepen:

```
@PROMPT $e[0;59;"HULP";13p

@PROMPT $p$g
```

Tip: Tussen beide opdrachten voegt u een witregel toe. ECHO mag niet uitgeschakeld zijn. Als de rest van uw bestand AUTOEXEC.BAT vereist dat ECHO is uitgeschakeld, zult u voor en na deze drie regels de ECHO in en weer uit moeten schakelen.

Sommige applicaties zullen niet goed meer werken als u toetsen een andere waarde geeft. Verwijder de opdrachten in dat geval uit het bestand AUTOEXEC.BAT en maak een apart batchbestand HELP_ON.BAT, waarin u de functie van F1 wijzigt. Een dergelijk bestand kan er als volgt uitzien:

```
@REM ************
@REM * F1: Help *
@REM ************
@PROMPT $e[0;59;"HULP";13p
@PROMPT $p$g
```

Voordat u het niveau van DOS verlaat en een applicatie start, zult u de oorspronkelijke functie van F1 weer moeten herstellen. Dat kan met het volgende batchbestand HELP_OFF.BAT:

```
@REM ******************************************
@REM * F1: oorspronkelijke functie herstellen *
@REM ******************************************
@PROMPT $e[0;59;0;59p
@PROMPT $p$g
```

Tip: We hebben de uitvoer van de opdrachten zelf naar het scherm verhinderd met een apestaart, omdat anders het herbenoemen van de functietoets niet werkt.

Toelichting

Wilt u een toets een andere waarde geven, dan dient u onderscheid te maken tussen de gewone toetsenbordcode en de uitgebreide toetsenbordcode. Dat heeft te maken met het feit dat DOS voor gewone toetsen een getal gebruikt (de ASCII-code), maar voor bijzondere toetsen een dubbele code: eerst een nul en dan een getal. Voor gewone toetsen ziet de opdracht er als volgt uit.

```
PROMPT $e[x;yp
```

'x' staat voor de ASCII-code van de toets en 'y' voor de nieuwe waarde, dus een teken of een string. Een teken geeft u daarbij eveneens op in ASCII-code, een string tussen aanhalingstekens. Ter verduidelijking de volgende voorbeelden:

PROMPT $e[90;48p^t	de toets 'Z' plaatst een nul op het scherm
PROMPT $e[90;"DIR"p	de toets 'Z' krijgt de betekenis 'DIR'
PROMPT $e[90;90p^t	de toets 'Z' krijgt zijn oorspronkelijke waarde terug

U mag bij 'y' ook meerdere waarden opgeven, gescheiden door puntkomma's. Zo kunt u bijvoorbeeld een opdracht direct uitvoeren, door de ASCII-code van Enter (13) toe te voegen. Voegt u aan het tweede voorbeeld hierboven ASCII-code 13 toe, dan is een druk op de toets Z voldoende voor een inhoudsopgave van een directory op uw scherm:

```
PROMPT $e[90;"DIR";13p
```

Uiteindelijk lenen de gewone toetsen zich niet zo best voor dit soort manipulaties. Het is erg vervelend als het systeem u hardnekkig directories laat zien, terwijl u gewoon een 'Z' wilt typen. De functietoetsen zijn hier wel bij uitstek voor geschikt.

4.7 De ASCII-tabel op het scherm

Een van de vele briefjes met geheugensteuntjes die rondom PC's te vinden zijn, bevat altijd een ASCII-tabel. Daarop kunt u bijvoorbeeld zien welke toetsencombinatie samen met de Alt-toets de Griekse letter Σ op het scherm tovert (ASCII-code 228). Met een klein programma in BASIC kunt u die tabel ook snel op uw scherm plaatsen.

Uitwerking

U zou met de hand een tekstbestand kunnen maken met in kolommen links de ASCII-code en rechts het bijbehorende teken. Dat tekstbestand kunt u dan met behulp van een batchbestand weergeven op uw scherm. Met een klein in BASIC geschreven programma gaat het invoeren sneller en makkelijker, zonder fouten. Het onderstaande programma ASCII.BAS maakt het tekstbestand ASCII.TXT aan en slaat het op in de directory C:\BAT.

```
REM Tekstbestand met ASCII-tabel aanmaken
OPEN "C:\BAT\ASCII.TXT" FOR OUTPUT AS #1
  FOR i = 32 TO 255 STEP 12
    FOR j = 0 TO 11
      IF i + j <= 99 THEN PRINT #1, " ";
      IF i + j < 256 THEN PRINT #1, i + j; CHR$(i + j);
    NEXT j
    PRINT #1, CHR$(13)
  NEXT i
CLOSE #1
END
```

Het batchbestand dat het zo aangemaakte tekstbestand op het scherm zet, ASCII.BAT, ziet er als volgt uit:

```
@REM ******************************************
@REM * ASCII-tabel uitvoeren naar het scherm *
@REM ******************************************
@ECHO OFF
CLS
COPY >NUL C:\BAT\ASCII.TXT CON
@ECHO ON
```

Start u dit batchbestand, dan verschijnt de ASCII-tabel op uw scherm zoals hieronder is afgebeeld.

```
 32     33 !   34 "   35 #   36 $   37 %   38 &   39 '   40 (   41 )   42 *   43 +
 44 ,   45 -   46 .   47 /   48 0   49 1   50 2   51 3   52 4   53 5   54 6   55 7
 56 8   57 9   58 :   59 ;   60 <   61 =   62 >   63 ?   64 @   65 A   66 B   67 C
 68 D   69 E   70 F   71 G   72 H   73 I   74 J   75 K   76 L   77 M   78 N   79 O
 80 P   81 Q   82 R   83 S   84 T   85 U   86 V   87 W   88 X   89 Y   90 Z   91 [
 92 \   93 ]   94 ^   95 _   96 `   97 a   98 b   99 c  100 d  101 e  102 f  103 g
104 h  105 i  106 j  107 k  108 l  109 m  110 n  111 o  112 p  113 q  114 r  115 s
116 t  117 u  118 v  119 w  120 x  121 y  122 z  123 {  124 ¦  125 }  126 ~  127 ⌂
128 Ç  129 ü  130 é  131 â  132 ä  133 à  134 å  135 ç  136 ê  137 ë  138 è  139 ï
140 î  141 ì  142 Ä  143 Å  144 É  145 æ  146 Æ  147 ô  148 ö  149 ò  150 û  151 ù
152 ÿ  153 Ö  154 Ü  155 ¢  156 £  157 ¥  158 ₧  159 ƒ  160 á  161 í  162 ó  163 ú
164 ñ  165 Ñ  166 ª  167 º  168 ¿  169 ⌐  170 ¬  171 ½  172 ¼  173 ¡  174 «  175 »
176 ░  177 ▒  178 ▓  179 │  180 ┤  181 ╡  182 ╢  183 ╖  184 ╕  185 ╣  186 ║  187 ╗
188 ╝  189 ╜  190 ╛  191 ┐  192 └  193 ┴  194 ┬  195 ├  196 ─  197 ┼  198 ╞  199 ╟
200 ╚  201 ╔  202 ╩  203 ╦  204 ╠  205 ═  206 ╬  207 ╧  208 ╨  209 ╤  210 ╥  211 ╙
212 ╘  213 ╒  214 ╓  215 ╫  216 ╪  217 ┘  218 ┌  219 █  220 ▄  221 ▌  222 ▐  223 ▀
224 α  225 ß  226 Γ  227 π  228 Σ  229 σ  230 µ  231 τ  232 Φ  233 Θ  234 Ω  235 δ
236 ∞  237 ø  238 ∈  239 ∩  240 ≡  241 ±  242 ≥  243 ≤  244 ⌠  245 ⌡  246 ÷  247 ≈
248 °  249 ∙  250 ·  251 √  252 ⁿ  253 ²  254 ■  255


C:\BAT>
C:\BAT>
```

ASCII-tabel

Toelichting

Het is ook bij dit programma belangrijk om de uitvoer van opdrachten te onderdrukken met ECHO OFF en de melding '1 file(s) copied' door de omleiding '>NUL'.

4.8 Snel diskettes formatteren

De mogelijkheden voor het formatteren van diskettes zijn in versie 5.0 aanzienlijk uitgebreid. Het proces is veiliger geworden en sneller als het om al eerder geformatteerde diskettes gaat. Toch kan het proces ook meer tijd in beslag nemen als MS-DOS eerst controleert of een diskette al is geformatteerd.

Uitwerking

Gebruikt u de juiste methode op het juiste moment, dan levert u dat nogal wat tijdwinst op. Er zijn in principe drie methoden:

Standaard — Als u de opdracht FORMAT geeft zonder parameters, controleert DOS eerst of de diskette al is geformatteerd. Is dat zo, dan controleert DOS de integriteit van de diskette, slaat de systeemgegevens van de diskette op en voorziet hem van nieuwe systeemgegevens. De methode duurt lang, vooral omdat alle sporen van de diskette worden gecontroleerd.

Permanent — In het Engels *unconditional*. De diskette wordt zonder meer geformatteerd. Als hij al geformatteerd is, gaan de gegevens erop voorgoed verloren.

Quick — U kunt alleen diskettes quick formatteren, als ze al eerder foutloos zijn geformatteerd. Het enige wat er gebeurt, is een controle van de capaciteit en het plaatsen van nieuwe systeemgegevens op de diskette.

Er zijn wel wat manieren om het formatteren van diskettes sneller te laten verlopen; ze komen er eigenlijk allemaal op neer overbodige handelingen over te slaan.

Nieuwe diskettes formatteren

Voeg aan de opdracht FORMAT de optie /u toe. Die voorkomt dat DOS eerst de capaciteit controleert. Het bespaart u bij het formatteren van een 5,25 inch-diskette met een capaciteit van 1,2 Mb ongeveer 25 seconden; het formatteren duurt in dit geval ongeveer 72 seconden.

Diskettes opnieuw formatteren en controleren

Geef de opdracht FORMAT zonder opties; er volgt dus een standaard-formattering, zoals hierboven is beschreven. Bij een diskette met een capaciteit van 1,2 Mb duurt dit proces ongeveer 52 seconden, toch weer zo'n 20 seconden winst ten opzichte van de vorige procedure. Bovendien houdt u de mogelijkheid de oude toestand weer te herstellen.

Quickformat van geformatteerde diskettes

Wilt u een al geformatteerde diskette eigenlijk wissen en gebruikt u daarvoor de opdracht FORMAT, voeg dan de optie /Q toe. Bij dit versnelde proces controleert DOS alleen de capaciteit van de diskette en schrijft de nieuwe systeemgegevens naar de diskette. Ook nu worden eerst de oude systeemgegevens van de diskette veilig opgeslagen. Bij een diskette met een capaciteit van 1,2 Mb duurt dit proces ongeveer 14 seconden.

Zo gaat DOS te werk

DOS onderneemt de volgende acties bij het formatteringsproces:

- DOS controleert het bestaande formaat.
- DOS slaat de oude systeemgegevens op.
- DOS controleert alle sectoren en sporen op integriteit.
- DOS schrijft de nieuwe systeemgegevens naar de diskette en geeft daarbij aan welke sectoren fouten bevatten en dus niet mogen worden gebruikt.

Het zou erg jammer zijn als u bij het formatteren van een gebruikte, maar foutloze diskette, de optie /Q zou vergeten. Maak daarom gebruik van batchbestanden en macro's om uw geheugen te ondersteunen. In beide gevallen voegt u de optie /Q toe. Zou u proberen een nieuwe diskette op deze manier te formatteren, dan merkt DOS dat en vraagt het systeem u of de diskette volgens de standaardmethode moet worden geformatteerd.

Formatteren via een batchbestand

Voor dit batchbestand maakt u gebruik van de gegevens uit paragraaf 4.1 en 4.2. Aan het daar besproken batchbestand voegt u eenvoudigweg de optie /Q toe. Noem het nieuwe batchbestand FORMAT.BAT.

```
FORMAT.COM %1 %2 %3 %4 %5 %6 %7 %8 %9 /Q
```

Definieer vervolgens met DOSKEY een macro om dit batchbestand te starten:

```
DOSKEY FORMAT=FORMAT.BAT $*
```

Deze macro zorgt ervoor dat niet het programma FORMAT.COM maar het batchbestand FORMAT.BAT wordt gestart. De optie $* zorgt voor het correct doorge-

ven van alle parameters van de opdrachtregel aan het batchbestand.

Formatteren via een macro

U kunt hetzelfde resultaat bereiken met behulp van een macro. In dat geval wordt het batchbestand FORMAT.BAT genegeerd. De macro luidt als volgt:

```
DOSKEY FORMAT=FORMAT.COM $* /Q
```

De macro roept het COM-programma FORMAT op en voegt aan de opgegeven parameters ($*) nog de optie /Q toe.

FORMAT starten zonder /Q

Bij verstek gebruikt u nu de snelle methode om te formatteren, maar er zijn natuurlijk altijd situaties waarop die niet van toepassing is. De optie /Q sluit het gebruik van de optie /U, om gegevens permanent te wissen, uit. U omzeilt dit probleem door het programma FORMAT rechtstreeks op te roepen, dus door toevoeging van de extensie .COM in de opdracht:

```
FORMAT.COM A: /U
```

U kunt ook de opdracht laten voorafgaan door een spatie:

```
<spatie>FORMAT.COM A: /U
```

Tip: Zie hoofdstuk 5 voor een uitgebreide bespreking van de mogelijkheden van DOSKEY.

4.9 DOS-omgevingsvariabelen instellen

MS-DOS is MS-DOS niet. Met deze paradox doelen we niet op de verschillende versies van MS-DOS, maar op de flexibiliteit van het besturingssysteem. Voor een groot deel komt die voor rekening van de systeemvariabelen of omgevingsvariabelen, met behulp waarvan u uw systeem zo configureert als het u het beste past.

Uitwerking

Geeft u de opdracht SET, dan krijgt u een overzicht op het scherm van de actuele instelling van de systeemvariabelen. Meestal gaat het om de volgende vier variabelen:

PROMPT De codering die aangeeft hoe de systeemprompt moet worden weergegeven.

PATH Het volledige zoekpad. Alle drives en directories die worden vermeld, worden doorzocht om bestanden te vinden die in een opdracht worden opgeroepen.

COMSPEC De naam en het zoekpad van het besturingssysteem. In bepaalde gevallen moet MS-DOS delen van het besturingssysteem opnieuw laden; op dat moment is de variabele belangrijk.

TEMP Een directory voor tijdelijke bestanden. DOS en diverse applicaties maken van de directory gebruikt om tijdelijk gegevens op te slaan. Dat is bijvoorbeeld het geval bij multi-tasking en bij opdrachten die gebruik maken van het *pipe*-symbool.

We noemen deze variabelen omgevingsvariabelen, omdat ze in zekere zin de omgeving bepalen waarbinnen MS-DOS werkt. Denk maar aan de menu's in applicaties waarmee u de weergave op het scherm kunt aanpassen.

Wijzigt u een systeemvariabele, dan zult u dat vroeger of later aan DOS merken. De voorgeschreven syntaxis van de opdracht is:

```
SET variabele=inhoud
```

Om de prompt te wijzigen kunt u dus de gewone DOS-opdracht invoeren of de systeemvariabele wijzigen:

```
SET PROMPT=$p $t$h$h$h$h$h$h$g
```

De bovenstaande prompt brengt behalve de drive en de actuele directory ook de actuele tijd in beeld. Zes maal '$h' verwijdert de seconden en honderdsten van seconden.

Systeemvariabelen op een zinvolle manier gebruiken

Het heeft weinig zin om achter PATH alle directories en subdirectories op te sommen van uw harde schijf. Een aantal directories kunt u echter vanaf het begin niet missen. Als u een tijdje aan het werk bent, zult u vanzelf merken dat u bepaalde wijzigingen in de variabelen aan moet brengen, soms alleen maar tijdelijk. De onderstaande instellingen zijn om mee te beginnen zeer bruikbaar.

variabele	inhoud
PROMPT	PROMPT pg
PATH	PATH C:\BAT;C:\DOS
COMSPEC	SET COMSPEC=C:\COMMAND.COM
TEMP	SET TEMP=C:\TEMP

Een prompt die tegelijkertijd aangeeft wat de actuele directory is, is uiterst zinvol: hij voorkomt bijvoorbeeld het wissen van bestanden in de verkeerde directory. Het zoekpad hoort altijd te leiden naar C:\BAT en C:\DOS (in die volgorde). Naarmate u meer applicaties gebruikt, zult u meer zoekpaden toevoegen (C:\WP;C:\DBASE;C:\WIN, enzovoort). Let erop dat het totale zoekpad niet te lang wordt (maximaal vijf à zes directories) en geef altijd ook de drive-letter op. Doet u dat niet, dan zal DOS alleen opdrachten en programma's vinden zolang C: de actuele drive is. Schakelt u om naar D: of A:, dan geeft DOS onmiddellijk de melding *Bad command or file name.*

COMSPEC wijzigen

Wijzig COMSPEC alleen als het besturingssysteem dat u gebruikt, zich niet in de hoofddirectory van de harde schijf bevindt of als u het systeem start vanaf diskette. In het eerste geval dient u het bestand CONFIG.SYS als volgt aan te passen:

```
SHELL=C:\DOS\COMMAND.COM /P
```

Aan het bestand AUTOEXEC.BAT voegt u een regel toe die aangeeft waar het besturingssysteem kan worden gevonden:

```
SET COMSPEC=C:\DOS\COMMAND.COM
```

Start u het systeem vanaf een diskette in drive A:, dan zal MS-DOS het besturingssysteem voortdurend zoeken in drive A:. Dat is lastig: u moet vaak diskettes verwis-

selen en het kost veel tijd. Als u in dat geval de systeembestanden van de diskette in drive A: kopieert naar een aparte directory op de harde schijf (niet C:\DOS!) en u stelt de systeemvariabele COMSPEC daarop in, kunt u rustig doorwerken. DOS zoekt het besturingssysteem in die directory.

Deze aanpassing heeft vooral zin als u met programma's werkt die een bepaalde versie van MS-DOS vereisen (zie ook paragraaf 3.4).

TEMP instellen

Stel deze omgevingsvariabele bij voorkeur in op een directory op de harde schijf, omdat de gegevensuitwisseling met de harde schijf veel sneller verloopt. Bovendien kunt u na een storing de bestanden makkelijker terugvinden. Maak dus op de harde schijf een directory aan:

```
MD C:\TEMP
```

Vervolgens stelt u de variabele in op deze directory:

```
SET TEMP=C:\TEMP
```

Een aantal applicaties (WORD, DBASE IV) verwacht een andere directory voor tijdelijke bestanden, zodat u het beste twee systeemvariabelen kunt definiëren:

```
SET TEMP=C:\TEMP
SET TMP=C:\TMP
```

Tip: Werkt u met een RAM-disk, dan kunt u daar uw directory TEMP op inrichten. Een RAM-disk werkt nog veel sneller dan een harde schijf. Het nadeel is dat u bij een storing al uw tijdelijke bestanden kwijt bent!

De systeemvariabele DIRCMD

De gegevens die u opslaat in de systeemvariabele DIRCMD gebruikt DOS als standaardmethode voor de weergave van directories op het scherm. Wilt u dat DOS bij de opdracht DIR de uitvoer altijd per pagina op het scherm zet, dan geeft u de variabele in het bestand AUTOEXEC.BAT de volgende inhoud:

```
SET DIRCMD=/P
```

Systeemvariabelen wijzigen in een batchbestand

In een aantal situaties kan het handig zijn systeemvariabelen vanuit een batchbestand te wijzigen. Als u bijvoorbeeld incidenteel een overzicht wilt van de bestanden in een directory inclusief de bestanden van de aanhangende subdirectories, kunt u gebruik maken van het volgende batchbestand. Daarin wordt niet alleen de systeemvariabele gewijzigd, maar ook weer ingesteld op de oorspronkelijke waarde:

```
SET TMP=%DIRCMD%
SET DIRCMD=
DIR /S
SET DIRCMD=%TMP%
```

U slaat dus de actuele inhoud van de systeemvariabele op in de tijdelijke variabele TMP. Daarna wist u de systeemvariabele (u geeft niets op achter het gelijkteken), geeft de eenmalige opdracht DIR /S en activeert ten slotte de systeemvariabele weer.

De merkwaardige schrijfwijze van de tijdelijke variabele %TMP% komt hieronder nog aan de orde. Deze notatie verwijst naar de letterlijke inhoud van de variabele.

Om te voorkomen dat uw harde schijf verstopt raakt met overbodige bestanden en bestandjes, kunt u nog een laatste opdracht toevoegen om de tijdelijke variabele te wissen:

```
SET TMP=
```

Een ander voorbeeld van het gebruik van batchbestanden om systeemvariabelen te wijzigen is het volgende, waarbij de opdracht PROMPT wordt gebruikt om de waarde van een functietoets aan te passen.

Tip: Het voorbeeld dat we hieronder bespreken, werkt alleen als u de device driver ANSI.SYS opneemt in het bestand CONFIG.SYS:

```
DEVICE=ANSI.SYS
```

Nadat u de opdracht hebt toegevoegd, dient u het systeem opnieuw te starten, voordat hij effect heeft.

Het volgende batchbestand, F10.BAT, roept de tekstverwerker WORD op met het bestand dat het laatst is bewerkt. De witregel in het batchbestand is belangrijk; hij zorgt ervoor dat de gewijzigde prompt een keer wordt uitgevoerd naar het scherm,

en dientengevolge ook wordt doorgegeven aan ANSI.SYS.

```
SET OLDPROMPT=%PROMPT%
PROMPT $e[0;68;"WORD /L";13p

SET PROMPT=%OLDPROMPT%
SET OLDPROMPT=
```

Hoewel het batchbestand de systeemprompt wijzigt, is die na beëindiging van het batchbestand weer in zijn oorspronkelijke staat hersteld. U kunt dat makkelijk controleren door een wat afwijkende systeemprompt te definiëren voordat u het batchbestand oproept.

4.10 Met eigen systeemvariabelen werken

Met behulp van de systeemvariabelen kunt u DOS makkelijk aanpassen aan uw eigen wensen. Daarnaast kunt u ook allerlei DOS-opdrachten onderbrengen in systeemvariabelen, zodat bijvoorbeeld de opdracht FORMAT automatisch wordt vertaald in FORMAT=A: /4.

Uitwerking

Zowel achter de systeemprompt als in batchbestanden kunt u een eigen systeemvariabele definiëren met behulp van de opdracht SET. U voert de naam van de opdracht in, gevolgd door een gelijkteken en de door u gekozen inhoud.

Zo kunt u bijvoorbeeld een variabele GEBRUIKER definiëren en laden met de waarde ‘Meier’. Dat gaat als volgt:

```
SET GEBRUIKER= Meier
```

Wilt u van alle systeemvariabelen alleen deze laatste bekijken, dan kunt u de opdracht SET (zonder parameters) filteren met de opdracht FIND. Gaan we uit van hetzelfde voorbeeld als hierboven, dan typt u:

```
SET | FIND "GEBRUIKER"
```

SET zoekt alle systeemvariabelen op, maar voert die niet uit naar het scherm, maar naar de opdracht FIND, die daaruit alleen de systeemvariabele ‘GEBRUIKER’ filtert.

Wilt u een systeemvariabele wissen, dan kunt u volstaan met een opdracht waarin u de naam van systeemvariabele laat volgen door een gelijkteken:

```
SET GEBRUIKER=
```

4.11 Systeemvariabelen gebruiken in batchbestanden

Nu u op de hoogte bent van het aanmaken, wijzigen en wissen van systeemvariabelen, willen we u uitleggen hoe u ze in batchbestanden kunt gebruiken om processen in MS-DOS te beïnvloeden.

Uitwerking

De bewerkingen van systeemvariabelen die we hierboven hebben besproken, kunt u zowel vanaf de systeemprompt als in batchbestanden uitvoeren. De inhoud van een systeemvariabele kunt u echter alleen in batchbestanden lezen en gebruiken.

Daartoe plaatst u de naam van de variabele tussen procenttekens. Om dus bijvoorbeeld in een batchbestand vast te stellen of de gebruiker Meier heet, zou u het volgende kleine batchbestand kunnen schrijven:

```
@ECHO De actuele gebruiker heet %GEBRUIKER%
```

Op het scherm zou dat het volgende opleveren:

```
De actuele gebruiker heet Meier
```

MS-DOS vervangt dus de tussen procenttekens ingesloten naam van de variabele door de waarde van de variabele. In tegenstelling tot wat bij veel programmeertalen gebruikelijk is, vervangt MS-DOS werkelijk de variabelenaam door de inhoud. De opdracht PRINT in BASIC voert bijvoorbeeld de inhoud van de variabele uit, terwijl in de opdracht zelf de naam van de variabele blijft staan. Dat dat zo is, kunt u zien als u de apestaart voor de ECHO-opdracht verwijdert. Op uw scherm komt dan te staan:

```
C:\>ECHO De actuele gebruiker heet Meier
De actuele gebruiker heet Meier
```

Dit levert interessante gebruiksmogelijkheden op: u kunt de inhoud van een systeemvariabele controleren en afhankelijk van de uitkomst de werkzaamheden een bepaalde kant opsturen.

We gaan ervan uit dat alleen de gebruiker Meier gebruik mag maken van de tekstverwerker WORD. De programmabestanden van WORD bevinden zich in de directory C:\WORD. In het zoekpad is C:\BAT opgenomen vóór C:\WORD. Het volgende batchbestand dient dan in C:\BAT te staan.

```
@ECHO OFF
IF NOT %GEBRUIKER%="Meier" GOTO Fout
C:\WORD\WORD %1 %2 %3
GOTO Einde
:Fout
ECHO U hebt geen toegang tot het programma
:Einde
ECHO ON
```

Als de naam Meier als gebruiker is opgegeven, roept het batchbestand WORD op en geeft het mogelijke parameters (%1, %2 en %3) door aan het programma. Als er echter een ander gebruiker is opgegeven, springt het batchbestand naar de foutmelding.

De heer Meier dient zich dus voordat hij begint te identificeren met de volgende opdracht:

```
SET GEBRUIKER= Meier
```

Die opdracht kan hij invoeren achter de systeemprompt of opnemen in een klein batchbestand. Als hij klaar is met zijn werk, dient hij natuurlijk de systeemvariabele weer te wissen.

Het is belangrijk het zoekpad exact aan te geven bij het oproepen van WORD, omdat het batchbestand anders voortdurend zichzelf zou oproepen en het systeem dus in een eindeloze lus terecht zou komen. In plaats van het zoekpad kunt u ook de extensie van het startbestand van WORD opgeven; in plaats van C:\WORD\WORD geeft u dan dus op:

```
WORD.EXE %1 %2 %3
```

4.12 De environment space uitbreiden

Als u veel systeemvariabelen gebruikt, is de geheugenruimte die MS-DOS daar bij verstek voor reserveert, niet toereikend. Er verschijnt dan een foutmelding op uw scherm:

```
Out of environment space
```

U kunt de geheugenruimte uitbreiden door het besturingssysteem te starten met bepaalde parameters.

Uitwerking

Bij verstek reserveert MS-DOS slechts 160 bytes voor de systeemvariabelen. Normaal gesproken is dat ook ruimschoots genoeg. Zodra u echter zelf meer variabelen definieert, bereikt u ook heel snel de kritische grens. Als u het besturingssysteem COMMAND.COM start met de optie /E:xxx, reserveert MS-DOS meer ruimte. Voert u de volgende opdracht in

```
COMMAND /E:500
```

dan laadt u in feite een tweede besturingssysteem, maar nu met meer gereserveerd geheugen voor de systeemvariabelen. Wilt u dit besturingssysteem weer verlaten, dan typt u achter de systeemprompt de opdracht:

```
EXIT
```

Als u permanent meer geheugenruimte voor uw systeemvariabelen wilt, kunt u in het bestand CONFIG.SYS al de nodige opdrachten plaatsen. U gebruikt daarvoor de opdracht SHELL met de naam van het besturingssysteem en de opties /P (permanent) en /E:

```
SHELL=C:\COMMAND.COM /P /E:500
```

Tip: De optie /P is nodig om het besturingssysteem permanent te laden, anders zou u na een EXIT-opdracht uw besturingssysteem beëindigen met als gevolg dat u de PC niet meer zou kunnen besturen (oftewel: het toetsenbord zou niets meer doen). Bovendien zorgt de optie ervoor dat het bestand AUTOEXEC.BAT wordt afgewerkt, zodat u over een prompt en een zoekpad beschikt.

4.13 Meerdere environments installeren

Het is u misschien al eens overkomen dat u bijvoorbeeld vanuit de DOS Shell een tweede besturingssysteem start en daarin systeemvariabelen definieert die later niet meer aanwezig blijken te zijn. Dat kan te maken hebben met de manier waarop applicaties het geheugen beheren.

Uitwerking

In MS-DOS kan een applicatie een ander programma laden en de besturing daaraan doorgeven. Als u dit tweede programma beëindigt, krijgt de eerste applicatie weer zeggenschap over het systeem. De DOS Shell functioneert net zo, als u een tweede besturingssysteem activeert.

Tip: Het probleem doet zich niet voor als u gebruik maakt van de *Task swapper* van de DOS Shell, en het tweede besturingssysteem niet verlaat met de opdracht EXIT maar via de Task swapper.

Ook de opdrachten Bibliotheek Systeem van WORD en de DOS Shell van Turbo Pascal 6.0 werken op deze manier.

De volgende twee uitgangspunten bepalen hoe een applicatie omgaat met de environment space:

- Het tweede programma dat wordt geladen, werkt binnen een kopie van de environment space van de eerste applicatie.
- Als dit tweede programma systeemvariabelen wijzigt, geldt dat alleen voor die kopie. Bij het beëindigen van dat programma geldt voor de oorspronkelijke applicatie nog steeds de oorspronkelijke environment space.

U kunt dit heel eenvoudig testen door een tweede besturingssysteem te laden. Dit komt overeen met het tweede programma, waarvan hierboven sprake is. Wijzig de systeemprompt in dat besturingssysteem en geef de opdracht SET. U ziet dan de gewijzigde systeemprompt.

Verlaat het tweede besturingssysteem weer met de opdracht EXIT en geef nogmaals de opdracht SET: de systeemprompt heeft nu weer het oorspronkelijke formaat gekregen.

MS-DOS-opdrachten gebruiken vanuit applicaties

Als u doorhebt hoe dit alles werkt, zult u zich niet meer verbazen over het feit dat applicaties bij tijd en wijle DOS-opdrachten gebruiken.

Zo kunt u vanuit WORD met de opdracht Bibliotheek Systeem tijdelijk naar DOS terugkeren, daar een ander besturingssysteem laden, allerlei bewerkingen uitvoeren en uiteindelijk via EXIT weer terugkeren naar WORD. Terwijl het tweede besturingssysteem actief is, kunt u naar hartelust systeemvariabelen wijzigen (bijvoorbeeld zoekpad en prompt) zonder dat dat effect heeft op het functioneren van WORD.

Als u vanuit WORD terugkeert naar DOS en een tweede besturingssysteem laadt, loopt u de kans dat de foutmelding *Out of environment space* zeer snel op uw scherm verschijnt. Het maakt niet uit of u in het oorspronkelijke besturingssysteem 500 bytes geheugenruimte voor systeemvariabelen hebt gereserveerd. Het tweede besturingssysteem wordt geladen met de default-geheugenruimte van 160 bytes. U zult bij het starten van het tweede besturingssysteem opnieuw meer geheugen moeten reserveren. Dat geldt ook als u vanuit Windows 3.0 vanuit de hoofdgroep de DOS-prompt oproept.

U zult dus het tweede besturingssysteem moeten oproepen met de optie /E:xxx. Als u in de DOS Shell werkt, markeert u daartoe in de *Program manager* de Command Prompt. Vervolgens opent u het menu File en kiest u Properties.... In het dialoogvenster voert u op de regel *Commands* achter COMMAND de optie /E:500 toe. Bij Windows 3.0 markeert u eerst het pictogram *DOS*. Vervolgens opent u het menu Bestand, kiest Kenmerken... en voert op de *Opdrachtregel* achter COMMAND.COM weer dezelfde optie in.

4.14 Een zoekpad vastleggen in een systeemvariabele

De opdrachtregel van MS-DOS bevordert de overzichtelijkheid, maar belemmert snel werken: hoe vaak gebeurt het niet dat u vanuit de directory waar u werkt, even naar een andere directory wilt en vervolgens weer terug. Via de opdracht CD moet u dan het hele zoekpad weer opnieuw invoeren. U kunt die actuele directory ook even vastleggen, zodat het terugkeren veel makkelijker wordt.

Uitwerking

U maakt daartoe een systeemvariabele CURRENTDIR. De waarde van de variabele is de actuele directory en u keert er steeds mee naar die directory terug.

Maak eerst een hulpbestand CD.DAT aan met de systeemvariabele:

```
@SET CURRENTDIR=
```

De systeemvariabele is in eerste instantie leeg. Sla het bestand zoals gewoonlijk op in de directory C:\BAT. Maak vervolgens het batchbestand CD_SET.BAT aan, dat de actuele directory als waarde toekent aan de systeemvariabele.

```
@REM CD_SET.BAT
@CD >C:\BAT\TEMP.$$$
@COPY /B C:\BAT\CD.DAT+C:\BAT\TEMP.$$$ C:\BAT\_CD.BAT >NUL
CALL C:\BAT\_CD.BAT
```

Met een tweede, qua omvang zeer bescheiden batchbestand, CD_GET.BAT, kunt u nu steeds terugkeren naar de actuele directory:

```
@REM CD_GET.BAT
CD %CURRENTDIR%
```

Deze procedure kunt u niet alleen vanaf de systeemprompt uitvoeren, maar ook vanuit een batchbestand. Net zo goed als u de systeemprompt tijdelijk kunt wijzigen met behulp van een batchbestand (en de tijdelijke variabele OLDPROMPT), kunt u de actuele directory tijdelijk wijzigen met de bovenstaande batchbestanden.

Het batchbestand is vooral erg nuttig als u een applicatie vanuit een bepaalde directory dient te starten, terwijl u na beëindiging van die applicatie toch weer in uw actuele directory wilt terugkeren. We zullen dat laten zien aan de hand van een voorbeeld waarin we dBASE IV starten vanuit de directory C:\DBASE met behulp van een batchbestand DBASE.BAT, dat we opslaan in C:\BAT.

```
@ECHO OFF
CALL CD_SET
C:
CD \DBASE
DBASE
CALL CD_GET
ECHO ON
```

Tip: Roep de beide batchbestanden CD_SET.BAT en CD_GET.BAT op met de opdracht CALL, anders keren ze na uitvoering van hun taak niet terug naar dit batchbestand.

4.15 Capaciteit van drives aanpassen met een systeemvariabele

Is uw PC uitgerust met een diskettestation met een capaciteit van 1,44 Mb en werkt u veel met diskettes van 720 Kb, dan dient u bij het formatteren steeds de juiste parameters aan te geven. U kunt daarvoor ook een systeemvariabele gebruiken.

Uitwerking

Definieer een systeemvariabele DR_A met de parameter en maak een batchbestand FORMAT.BAT aan met de volgende opdrachten:

```
@ECHO OFF
FORMAT A: %DR_A%
ECHO ON
```

Tip: Gebruikt u al een batchbestand FORMAT.BAT voor andere doeleinden, dan voegt u de opdrachten die hierboven staan, toe aan dat bestand.

Wilt u in drive A: een diskette formatteren met een capaciteit van 1,44 Mb, dan wist u eenvoudigweg de systeemvariabele. U kunt deze ruil van parameters ook onderbrengen in twee batchbestanden, 720.BAT en 144.BAT. 720.BAT ziet er dan als volgt uit:

```
@REM 720.BAT
SET DR_A=/F:720
```

Als deze constructie u bevalt, kunt u die uiteraard nog uitbreiden met tal van andere batchbestanden om DOS zo veel mogelijk aan uw wensen aan te passen. Voor vrijwel alle randapparaten mag u systeemvariabelen definiëren en die vervolgens met batchbestanden aansturen. Zo kunt u bijvoorbeeld met behulp van een batchbestand reservekopieën van bestanden naar een diskette schrijven, waarbij u de directory voor die bestanden vastlegt in een systeemvariabele. Met de volgende opdracht kopieert u alle nieuwe bestanden uit de directory C:\GEGEVENS inclusief subdirectories naar A:

```
XCOPY C:\GEGEVENS A: /S /M
```

In plaats daarvan kunt u ook de volgende opdracht invoeren:

```
SET RESDIR=C:\GEGEVENS
```

HIermee kunt u de directory opgeven die in het batchbestand RESKOP.BAT moet worden gebruikt. RESKOP.BAT ziet er als volgt uit:

```
@REM RESKOP.BAT
XCOPY %RESDIR% A:
```

Dit basisidee kunt u weer verwerken in andere procedures, bijvoorbeeld in een batchbestand waarin u de tekstverwerker WORD start vanuit verschillende directories, afhankelijk van de gebruiker:

```
@REM WORD.BAT
IF "%Gebruiker%"=="Meier" CD \TEKSTEN\MEIER
IF "%Gebruiker%"=="Scholte" CD \TEKSTEN\SCHOLTE
IF "%Gebruiker%"=="" CD \WORD
WORD %1 %2 %3
```

4.16 Het laatst gebruikte gegevensbestand vastleggen in een systeemvariabele

In de laatste paragraaf van dit hoofdstuk willen we u laten zien hoe u allerlei instellingen van programma's kunt bewaren met behulp van systeemvariabelen. WORD 5.0 kent een functie waarmee u automatisch het bestand laadt waar u het laatst in hebt gewerkt: bij het starten van het programma voegt u dan de optie /L toe. De meeste andere applicaties beschikken niet over die functie. Met behulp van systeemvariabelen kunt u dit probleem oplossen.

Uitwerking

Voor het programma WORKS zou u een systeemvariabele WORKS kunnen definiëren, die als waarde de naam van het laatst bewerkte bestand krijgt. U start daarna WORKS met een batchbestand WORKS.BAT die de systeemvariabele doorgeeft aan het programma:

```
C:\WORKS\WORKS %WORKS%
```

Van het bewerkte bestand kunt u na beëindiging van het programma met behulp van dezelfde systeemvariabele een reservekopie op een diskette maken:

```
COPY %WORKS% A:
```

U zult voor verschillende applicaties eigen oplossingen moeten zoeken om de naam van het actuele bestand vast te leggen. In WORKS 2.0 kunt u een macro maken, die via de opdracht BEWERKEN/EXTRA INVOEGEN de bestandsnaam in een opdracht SET WORKS=*bestandsnaam* plaatst en de bestandsnaam daarna via de printer driver TTY opslaat in een bestand. Dat bestand kunt u dan samen met WORKS laden.

Natuurlijk kunt u WORKS ook op de gebruikelijke manier met de bestandsnaam starten:

```
WORKS TABEL.WKS
```

Gebruikt u deze procedure vaak, dan kunt u de oproep van het programma weer onderbrengen in een batchbestand, dat de naam van het gegevensbestand kent:

```
@REM WORKS.BAT
ECHO OFF
IF "%1%"=="/l" GOTO Laatste
IF "%1%"=="/L" GOTO Laatste
SET WORKS=%1
:Laatste
WORKS.EXE %WORKS% %2 %3
IF NOT "%WORKS%"=="" COPY %WORKS% A:
```

Het batchbestand controleert eerst of de optie /L aan de oproep van WORKS is toegevoegd. Zo ja, dan slaat het batchbestand het opnieuw instellen van de systeemvariabele over. Zo nee, dan wordt de systeemvariabele geladen met de eerste parameter die is meegegeven aan de oproep. Roept u WORKS bijvoorbeeld als volgt op:

```
WORKS TABEL.WKS
```

dan wordt de eerste variabele %1 geladen met de waarde 'TABEL.WKS'. Roept u WORKS op zonder parameters, dan wordt uiteindelijk WORKS gestart zonder dat een gegevensbestand wordt geladen.

Geeft u bij de oproep de optie /l of /L aan (batchbestanden maken onderscheid tus-

sen hoofdletters en kleine letters), dan wordt WORKS geladen met het bestand waarvan de naam in de systeemvariabele is opgeslagen.

Tip: Omdat systeemvariabelen verloren gaan als u de PC uitschakelt, werkt het bovenstaande batchbestand alleen tijdens de sessie waarin u de systeemvariabele een bepaalde waarde hebt toegekend.

5 Tips en trucs voor het invoeren van opdrachten

Bij discussies over de kwaliteiten van MS-DOS wijzen tegenstanders van het systeem steevast naar de moeizame manier van opdrachten invoeren via de opdrachtregel: tientallen verschillende opdrachten, een veelvoud daarvan aan parameters en opties, en dat alles in een nauw omschreven syntaxis. Het alternatief, bijvoorbeeld een moderne grafische gebruikersinterface, wekt vooral bij gevorderde gebruikers van DOS nogal eens weerstand op, wegens zijn al of niet vermeende beperkingen en omslachtigheid: menu's openen, dialoogvensters invullen, controlevragen beantwoorden en dat alles voor één simpele opdracht...

Het laatste woord in die discussie zal wel nooit gesproken worden. Feit is dat versie 5.0 van MS-DOS voor het invoeren van opdrachten belangrijk is verbeterd. In dit hoofdstuk zullen we die verbeteringen bespreken.

5.1 Het gebruikersvriendelijke DOSKEY

Tot nu toe was een enkel typefoutje bij het invoeren van een opdracht achter de systeemprompt voldoende om een foutmelding te veroorzaken. Meestal zonder kwalijke gevolgen voor het systeem, maar wel vervelend voor u als gebruiker: u moest de opdracht helemaal opnieuw typen. Correcties in de opdrachtregel waren niet mogelijk. Het hulpprogramma DOSKEY helpt u van deze narigheid af, als u het op de juiste manier installeert.

Uitwerking

U dient het programma DOSKEY zo te installeren dat het vanaf de systeemstart actief is. Dat kan het beste met behulp van het bestand AUTOEXEC.BAT. Beschikt uw PC over een 386-processor en maakt u gebruik van het *Upper Memory* via EMM386.EXE, dan kunt u DOSKEY installeren zonder verlies van werkgeheugen met de opdracht:

```
LOADHIGH DOSKEY
```

Maar ook op andere machines die minder geavanceerd zijn, loont het de moeite vier Kb geheugenruimte te reserveren voor het installeren van DOSKEY. Er zullen wei-

nig applicaties zijn die net niet meer willen draaien vanwege die vier Kb. Tegelijkertijd bespaart u zichzelf wel veel ergernis, omdat u op de opdrachtregel kunt werken alsof het tekstverwerker is.

De belangrijkste toetsen van DOSKEY zijn:

Cursor omhoog	plaatst de vorige opdracht achter de systeemprompt
Cursor omlaag	plaatst een volgende opdracht op het scherm, tenzij de actuele opdracht de laatste in de buffer is
Cursor naar links	verplaatst de cursor een teken naar links
Cursor naar rechts	verplaatst de cursor een teken naar rechts
Del	wist teken boven de cursor
Ins	omschakeling tussen invoeg- en overschrijfmodus; in de overschrijfmodus werkt de Backspace-toets als cursor naar links
Backspace	wist in de invoegmodus het teken links van de cursor

Tip: De functietoetsen waarmee u tot versie 5.0 de opdrachtregel op een primitieve manier kon bewerken, zijn gehandhaafd. F3 heeft dus dezelfde functie als cursor omhoog, zij het dat u daar alleen de vorige opdracht mee terug kunt halen.

Opdrachten bewerken

Wilt u een opdracht corrigeren, dan roept u die opdracht eerst met cursor omhoog op achter de systeemprompt. Vervolgens plaatst u de cursor met behulp van de toetsen cursor naar links en cursor naar rechts onder de tekens die u wilt wijzigen. Bij verstek is DOSKEY ingesteld op de overschrijfmodus, zodat u nu de juiste tekens over de oude heen kunt typen. Met Del verwijdert u tekens; met Ins schakelt u om naar de invoegmodus om ruimte te maken voor extra tekens en Backspace te gebruiken. De cursor verandert van een streepje in een blokje.

Naast de toetsen die hierboven al zijn beschreven, kent DOSKEY nog een aantal toetsen die bij tekstverwerkers worden gebruikt om de cursor te verplaatsen en correcties aan te brengen.

Met de Home-toets brengt u de cursor in één keer naar het begin van de opdrachtregel.

Drukt u op de Esc-toets, dan wordt de opdracht achter de systeemprompt gewist. Dat is erg handig als u met cursor omhoog een oude opdracht hebt opgehaald uit de buffer, maar u besluit dat het toch gemakkelijker is de opdracht opnieuw in te voeren, omdat u te veel moet wijzigen. Gaat het inderdaad om een opdracht die u al eerder hebt afgesloten met Enter, dan blijft die wel in de buffer staan.

Gecombineerd met de Ctrl-toets verplaatsen de toetsen cursor naar links en cursor naar rechts de cursor niet één teken, maar één woord. Staat de typefout midden in de opdracht aan het begin van een woord of parameter, dan is dat de snelste manier om de cursor er naar toe te brengen. De opdracht 'XCOPY V:\TEKSTEN\BRIEVEN*.* A: /S /M' corrigeert u zo snel na de foutmelding *Invalid drive specification.* U haalt de opdracht uit de buffer met cursor omhoog en houdt de Ctrl-toets ingedrukt, terwijl u vier keer op cursor naar links drukt. De cursor staat nu onder de 'V' en u kunt de juiste drive-letter typen.

Gebruik de toetsencombinatie Ctrl-Home om alle tekens vanaf de plaats van de cursor naar het begin van de opdracht te wissen. Ctrl-End heeft het omgekeerde effect: alle tekens vanaf de cursor tot het einde van de opdracht worden gewist.

Hieronder geven we nog even een overzicht van deze toetsen.

Home	verplaatst de cursor naar het begin van de regel
End	verplaatst de cursor naar het einde van de regel
Esc	wist de opdracht achter de systeemprompt
Ctrl-cursor naar links	verplaatst de cursor één woord naar links
Ctrl-cursor naar rechts	verplaatst de cursor één woord naar rechts
Ctrl-Home	wist alle tekens links van de cursor
Ctrl-End	wist alle tekens rechts van de cursor

Opdrachten zoeken in de buffer

De buffer van DOSKEY kan zonder problemen enige tientallen opdrachten bevat-

ten, afhankelijk van hun omvang. Als u er daarvan één nodig hebt, kunt u met cursor omhoog net zo lang zoeken tot u hem hebt gevonden. Er zijn echter een paar speciale functies die dit soort werk vergemakkelijken. We zullen die functies bespreken aan de hand van een paar voorbeelden.

Zoeken met F8

We gaan ervan uit dat u de volgende opdrachten hebt gegeven:

```
COPY C:\DOS5\*.COM A:
DIR A:
CHKDSK A:
```

Daarna hebt u een aantal andere handelingen verricht. Nu wilt u dezelfde kopieerslag nog een keer uitvoeren. U kunt met cursor omhoog zoeken naar de opdrachten, maar u kunt ook het begin van de opdracht typen en vervolgens op F8 drukken. DOSKEY zoekt in de buffer de eerste opdracht die bij de zoekstring past en plaatst die op de opdrachtregel. In ons voorbeeld zou u het volgende kunnen doen:

```
CO <F8>
```

Krijgt u een andere opdracht te zien dan de gezochte, omdat u nog meer bestanden hebt gekopieerd, dan drukt u net zo lang op F8 tot de juiste opdracht achter de systeemprompt verschijnt. DOSKEY blijft zoeken.

Een overzicht van opdrachten met F7

In het vorige voorbeeld zijn we ervan uitgegaan dat u nog wist welke opdracht u had gebruikt. Als u niet meer weet of dat nu COPY of XCOPY was, kunt u met F7 een overzicht van alle opdrachten in de buffer op uw scherm laten weergeven. Het overzicht wordt per pagina weergegeven; druk op de spatiebalk om een volgende pagina te zien. De opdrachten zijn genummerd. Onthoud het nummer van de gezochte opdracht en breek de functie af met Ctrl-C.

Een opdracht terughalen met F9

Drukt u vervolgens op F9, dan vraagt DOSKEY u om het nummer van de opdracht met *Line number:*. U typt het nummer en sluit af met Enter. De opdracht wordt achter de systeemprompt gezet.

Hieronder geven we nog een overzicht van deze functies.

PgUp plaatst de eerste opdracht in de buffer op uw scherm

PgDn plaatst de laatste opdracht in de buffer op uw scherm

F7 plaatst een overzicht van alle opdrachten in de buffer per pagina op uw scherm

F8 zoekt een opdracht waarvan u de eerste tekens hebt getypt en plaatst die achter de systeemprompt

F9 haalt de opdracht waarvan u het nummer opgeeft, uit de buffer

Problemen met DOSKEY

Start u DOSKEY niet vanaf de systeemprompt, maar vanuit applicaties of bijvoorbeeld vanuit de DOS Shell, dan kunnen er vreemde dingen gebeuren: er verschijnen merkwaardige meldingen op uw scherm of DOSKEY werkt zelfs helemaal niet. Dit heeft alles te maken met het moment waarop u DOSKEY installeert.

Uitwerking

U dient DOSKEY te installeren voordat u applicaties en programma's als DOS Shell of Windows start. Alleen in dat geval werkt het programma foutloos.

Als u DOSKEY vanuit DOS Shell of Windows start, zal afhankelijk van de omstandigheden het volgende gebeuren:

- U start DOSKEY vanuit de DOS Shell (met File en Run) of vanuit Windows (met een dubbelklik op de bestandsnaam in het venster van Bestandsbeheer). Het programma wordt geladen en het systeem meldt *DOSKey installed.*. Vervolgens wordt de besturing echter weer teruggegeven aan DOS Shell of Windows.
- U laadt vanuit DOS Shell of Windows een tweede besturingssysteem (COMMAND.COM) en roept daarin DOSKEY op. U kunt nu met het programma werken, tot u het tweede besturingssysteem verlaat met EXIT. Roept u vervolgens het tweede besturingssysteem nogmaals op, dan zult u DOSKEY opnieuw moeten installeren.

In alle gevallen blijft het het beste DOSKEY te installeren via het bestand AUTOEXEC.BAT.

Toelichting

Dat DOSKEY zich zo gedraagt, heeft te maken met de aard van het programma. DOSKEY is een zogenaamd TSR-programma (*Terminate and Stay Resident*): als u het oproept, wordt het in het geheugen geïnstalleerd, maar onmiddellijk ook weer beëindigd. Het blijft echter in het geheugen, zolang u de computer niet uitschakelt, en leidt daar een min of meer slapend bestaan. Ondertussen houdt DOSKEY echter de DOS-interrupt in de gaten die verantwoordelijk is voor de invoer van opdrachten. Steeds als u de opdrachtregel activeert, vangt DOSKEY de actie op en doet daar iets mee, voordat hij de actie doorgeeft aan de rest van het systeem.

Start u DOSKEY vanuit een besturingssysteem dat u daarna weer verlaat met EXIT, dan verdwijnt niet alleen dat besturingssysteem, maar ook de daaraan gekoppelde kopie van DOSKEY.

5.2 Macro's maken met behulp van DOSKEY

Met de introductie van DOSKEY is het mogelijk geworden op eenvoudige wijze omvangrijke macro's te schrijven onder MS-DOS.

Uitwerking

De macro's die u met DOSKEY maakt, zijn in feite een soort batchbestanden, omdat u een aantal opdrachten koppelt aan een naam. U gaat als volgt te werk om bijvoorbeeld een macro te maken die een overzicht van de inhoud van een directory op het scherm plaatst, gesorteerd op bestandsnaam. De macro zorgt ervoor dat u de desbetreffende optie /o:n niet hoeft te onthouden. We noemen de macro DIRS:

```
DOSKEY DIRS=DIR /O:N
```

Wilt u nu een op naam gesorteerd overzicht van bestanden, dan voert u eenvoudigweg de opdracht DIRS in; u hebt eigenlijk zelf een nieuwe DOS-opdracht ontworpen.

Een macro maakt u dus door DOSKEY op te roepen, gevolgd door de naam van de macro, een gelijkteken en de inhoud van de opdracht. Wilt u meerdere opdrachten aan de macro toevoegen, dan scheidt u die met het teken *$t*. De macro DISK formatteert en test een diskette:

```
DOSKEY DISK=FORMAT A: $t CHKDSK A:
```

Er zijn veel overeenkomsten tussen batchbestanden en macro's die u met DOSKEY maakt. Er zijn echter ook een paar belangrijke verschillen:

1 In batchbestanden kunt u variabelen opnemen (%1 %2, enzovoort) voor het gebruik van parameters in de opdrachtregel. In DOSKEY-macro's gebruikt u daarvoor de codering $1, $2, enzovoort.

2 U kunt zelfs interne DOS-opdrachten vervangen door DOSKEY-macro's. MS-DOS geeft macro's voorrang, zelfs boven interne DOS-opdrachten. Maakt u een macro DIR, dan zal DOS die uitvoeren in plaats van de eigen interne opdracht.

3 Macro's werken sneller dan batchbestanden, omdat het systeem geen bestand hoeft te openen.

4 U kunt een macro oproepen, onafhankelijk van de drive of de directory die actief is; macro's werken onafhankelijk van het zoekpad.

5.3 Voorbeeld: drives activeren, een macro om rampen te voorkomen

Misschien is het u al een keer overkomen: u wilt alle bestanden van een diskette in drive A: wissen. U plaatst de diskette in de drive, typt per ongeluk 'A' in plaats van 'A:', drukt op Enter en typt zonder naar het scherm te kijken DEL *.*. Het systeem heeft ondertussen vastgesteld dat 'A' geen geldige opdracht is, geeft een foutmelding, maar blijft actief in de actuele directory en voert daar onbekommerd de wel geldige opdracht DEL *.* uit.

De onderstaande macro helpt u uit de brand. Definieer de macro onder de naam A en geef hem een inhoud mee die wel de drive A: activeert. Voeg er voor alle zekerheid de uitvoer van een geluidssignaal aan toe om u erop opmerkzaam te maken dat er iets bijzonders is gebeurd:

```
DOSKEY A=A: $t ECHO ^G
```

Tip: Typ bij het invoeren van de macro eerst de opdracht ECHO, druk vervolgens op Ctrl-G; op het scherm komt '^G' te staan en bij uitvoering van de macro klinkt het geluidssignaal.

Vanzelfsprekend is het verstandig voor iedere drive van uw PC een dergelijke macro aan te maken.

Als u zo verstandig bent om voortdurend reservekopieën te maken van gegevensbestanden, heeft het zin de opdracht XCOPY met parameters onder te brengen in de volgende macro RESKOP:

```
DOSKEY RESKOP=XCOPY *.* A: /M
```

U kunt zich dan niet vergissen in de optie die nodig is om alleen gewijzigde bestanden te kopiëren. De optie /M controleert het zogenaamde *Archive Attribute*. Dat wordt gezet als u wijzigingen in een bestand aanbrengt. Nadat ze zijn gekopieerd met XCOPY, wordt het attribuut teruggezet. Start u de macro twee keer achter elkaar, dan wordt in tweede instantie geen enkel bestand meer gekopieerd.

5.4 Variabelen gebruiken in macro's

De macro uit de vorige paragraaf waarmee u een reservekopie van gewijzigde bestanden maakt, is beperkt: u kunt er alleen mee naar A: kopiëren. DOSKEY staat echter het gebruik van variabelen toe, zodat u de macro kunt starten met opgave van parameters. U gebruikt voor de variabelen in macro's niet het procentteken, maar het dollarteken.

Voorbeeld: diskettes formatteren

De macro DISK hieronder kunt u gebruiken om in een willekeurige drive een diskette te formatteren en vervolgens te testen:

```
DOSKEY DISK=FORMAT $1 $t CHKDSK $1
```

Tip: De tekens '$t' scheiden de beide opdrachten in de macro. Enter als scheidingsopdracht is niet mogelijk, omdat u de macro afsluit met Enter. Op de scheidingstekens die u in macro's mag gebruiken, komen we hieronder nog terug.

U start de macro vervolgens door de naam ervan te typen en de drive-letter toe te voegen als parameter:

```
DISK A:
```

Wilt u meer variabelen opnemen in de macro, dan gebruikt u de plaatshouders $2, $3, enzovoort. In batchbestanden heeft de variabele %0 een heel speciale betekenis: hij verwijst altijd naar de naam van het batchbestand zelf. Voor macro's zou dus

nogal onzinnig zijn, zodat $0 in de syntaxis van macro's dan ook geen waarde heeft. Macro's kennen wel de speciale variabele $*, die alle parameters opneemt, ongeacht de spaties waardoor de parameters in de opdrachtregel worden gescheiden.

Voorbeeld: directories wissen

Het blijft een tamelijk omslachtig karwei om een directory te wissen, waarin zich nog bestanden bevinden. U kunt daarvoor een batchbestand aanmaken, maar ook een macro DELD (*DEL Directory*):

```
DOSKEY DELD=DEL $1 $t RD $1
```

Om alle handelingen uit te voeren hoeft u nog slechts de macro op te roepen met de naam van de directory als parameter.

Let op: Wissen blijft gevaarlijk! Geeft u bij deze macro niet de naam van een directory op, maar van een bestand, dan wordt dat bestand gewist. De macro vervangt eenvoudigweg de variabele door de parameter, zonder te controleren waar die parameter voor staat.

Bijzondere tekens gebruiken in macro's

Een aantal tekens heeft op de opdrachtregel een bijzondere status: ze worden door het besturingssysteem onmiddellijk geïnterpreteerd als opdrachten en uitgevoerd. Het zijn in de eerste plaats de tekens '<' en '>' voor het omleiden van in- en uitvoer, en het pipe-symbool waarmee u opdrachten koppelt. Zou u de tekens zo gebruiken, dan zou het besturingssysteem ze verkeerd gebruiken.

In DOSKEY worden deze tekens vervangen door andere:

<	Gebruik om invoer om te leiden de tekens $L.
>	Gebruik $G om uitvoer om te leiden naar een bestand of naar de printer. Twee keer $G achter elkaar vervangt de opdracht >> waarmee u de omgeleide uitvoer aan het einde van een bestand toevoegt.
/	Gebruik voor het pipe-symbool de combinatie $B.
$	Omdat het dollarteken zelf wordt gebruikt voor variabelen, kan het niet

zonder meer in een zelfnoemfunctie worden gebruikt. Verdubbel het dollarteken in dat geval.

Enter U kunt in macro's opdrachten niet, zoals in batchbestanden, van elkaar scheiden met Enter. Enter sluit de macro af; gebruik in plaats daarvan $t als scheidingsteken tussen twee opdrachten.

Om een overzicht van de inhoud van een directory om te leiden naar de printer kunt u dus de volgende macro gebruiken:

```
DOSKEY PRINTDIR=DIR $G LPT1:
```

Bij uitvoering wordt deze opdracht geïnterpreteerd als 'DIR > LPT1:'.

Een universele macro

Als u optimaal gebruik maakt van parameters, kunt u zelfs een macro aanmaken waarmee u vrijwel alle kanten op kunt. We laten dat zien aan de hand van de macro DRUK, die de resultaten van een opdracht naar de printer stuurt:

```
DOSKEY DRUK=$1 $2 $3 $4 $5 $G LPT1
```

Als u met de opdracht DIR alle bestanden met de extensie .EXE in de directory C:\DOS verdeeld over vijf kolommen op het scherm wilt zien, geeft u de opdracht:

```
DIR C:\DOS\*.EXE /W
```

Wilt u de uitvoer van die opdracht naar de printer sturen, dan gebruikt u de macro:

```
DRUK DIR C:\DOS\*.EXE /W
```

Bij het werken met DOSKEY dient u zich wel te realiseren dat het programma nauw verbonden is met de DOS-opdrachtregel. In applicaties (tekstverwerkingsprogramma's, databases) werkt DOSKEY niet. Evenmin kunt u DOSKEY-macro's activeren vanuit batchbestanden. Dat laatste is overigens geen al te groot bezwaar, omdat alles wat u met macro's kunt doen ook in batchbestanden mogelijk is.

5.5 Macro's opslaan

Macro's zijn in principe tijdelijk van aard: schakelt u de computer uit en weer in, dan gaan uw macro's verloren. U kunt macro's echter vanuit batchbestanden definiëren, zodat ze toch bewaard blijven.

Uitwerking

De macro's die we hiervoor hebben besproken, kunt u bestendig maken door ze op te nemen in het bestand AUTOEXEC.BAT:

```
DOSKEY DRUK=$1 $2 $3 $4 $5 $G LPT1:
DOSKEY DIRS=DIR /O:N
...
```

Wilt u de macro's niet automatisch bij iedere systeemstart definiëren, dan kunt u ook een apart batchbestand maken, bijvoorbeeld onder de naam MACRO.BAT. U roept dan het batchbestand op als u de macro's wilt gebruiken. Het is daarbij niet van belang of DOSKEY op dat moment al is geïnstalleerd.

Macro's omleiden naar een batchbestand

U hebt een paar goed functionerende macro's gemaakt met DOSKEY en wilt ze nu opslaan in een batchbestand MACRO.BAT. U hoeft de macro's niet opnieuw te typen: ze zijn in het geheugen aanwezig en u kunt gebruik maken van de parameter /MACROS om ze vanuit de buffer in het batchbestand te laden. Aansluitend hoeft u in het batchbestand alleen nog voor de macrodefinities de opdracht DOSKEY te plaatsen.

We gaan ervan uit dat het bestand MACRO.BAT nog niet bestaat. Met de opdracht:

```
DOSKEY /MACROS >MACRO.BAT
```

schrijft u alle macro-definities naar het batchbestand. In ons geval ziet het batchbestand er dan als volgt uit:

```
DRUK=$1 $2 $3 $4 $5 $G LPT1:
DIRS=DIR /O:N
```

Alleen de opdracht DOSKEY ontbreekt nog op beide regels. Start de editor EDIT

met de naam van het batchbestand als parameter:

```
EDIT MACRO.BAT
```

Voeg op beide regels 'DOSKEY' toe. Nadat u het bestand hebt opgeslagen en de editor hebt verlaten, kunt u de macro's beschikbaar maken door het batchbestand op te roepen.

Tip: Gaat het om veel opdrachten, dan kunt u het invoegen van het woord 'DOSKEY' in EDIT op zich weer enigszins automatiseren. U typt het woord een keer, markeert het en drukt op Ctrl-Ins om het op te slaan in een tijdelijk bestand (het *Clipboard*). Verplaats de cursor naar het begin van de volgende regel en druk op Shift-Ins om het gemarkeerde woord in te voegen. De cursor blijft aan het begin van de regel staan. Druk dus een keer op cursor omlaag en voeg het woord weer in met Shift-Ins. Herhaal dit zo vaak als nodig is.

5.6 Batchbestanden aanmaken met behulp van DOSKEY

Doskey slaat alle opdrachten die u invoert op de opdrachtregel, op in een buffer. U kunt van die buffer gebruik maken om de opdrachten op te slaan in een batchbestand.

Uitwerking

U gaat als volgt te werk:

1 Wis de buffer met Alt-F7.

2 Voer successievelijk vanaf de systeemprompt de opdrachten uit die samen het batchbestand moeten vormen. DOSKEY slaat ze stuk voor stuk op in de buffer.

3 Maak van de serie opdrachten een batchbestand met de volgende opdracht:

```
DOSKEY /HISTORY >WERK.BAT
```

De optie /HISTORY zorgt ervoor dat DOSKEY alle tot nu toe ingevoerde opdrachten zonder de regelnummers uitvoert. Omdat u de uitvoer hebt omgeleid naar een

bestand WERK.BAT, is in principe het batchbestand al klaar. DOSKEY is echter zeer consequent: ook deze laatste opdracht is opgenomen in de buffer en dus in het batchbestand! Roep dus aansluitend EDIT op met WERK.BAT als invoerbestand en verwijder de laatste regel.

Tip: Laat u er niet van weerhouden op de hierboven beschreven manier een batchbestand aan te maken als u pas na het uitvoeren van een aantal opdrachten op dat idee komt. In die situatie hebt u de buffer niet eerst met Alt-F7 gewist en dus zal er veel ‘rommel’ in het batchbestand terechtkomen. Het is gemakkelijker om met EDIT het bestand op te schonen dan een hele serie opdrachten opnieuw in te voeren.

Hulpprogramma’s voor het maken van batchbestanden

Het is niet helemaal bevredigend om na het maken van het batchbestand nog een keer met behulp van EDIT wijzigingen aan te moeten brengen. De beide onderstaande programma’s, het QBASIC-programma VERWRGL.BAS en het batchbestand MAKEBAT.BAT nemen dat werk voor het grootste deel van u over.

Uitwerking

U maakt het batchbestand in drie stappen:

1 Wis de buffer van DOSKEY met Alt-F7.

2 Voer de reeks opdrachten in die samen het batchbestand moeten vormen, bijvoorbeeld:

```
FORMAT A:
CHKDSK A:
DIR A:
```

3 Start het batchbestand MAKEBAT.BAT. Laat daarbij de extensie achterwege, bijvoorbeeld (aansluitend op punt 2):

```
MAKEBAT DISK
```

Het resultaat van het voorbeeld is een batchbestand DISK.BAT waarmee u eerst een diskette formatteert, hem vervolgens test en ten slotte een (leeg) overzicht van de inhoud op het scherm laat weergeven.

Het belangrijkste van de twee programma's die we gebruiken, is het QBASIC-programma VERWRGL.BAS. Het programma leest een bestand BRON.MAK in de directory C:\TEMP en maakt op grond van de inhoud daarvan een bestand DOEL.MAK aan in dezelfde directory. Het enige verschil tussen beide bestanden is het ontbreken van de laatste regel in het bestand DOEL.MAK:

```
REM De laatste regel van een bestand verwijderen
OPEN "C:\TEMP\BRON.MAK" FOR INPUT AS #1
OPEN "C:\TEMP\DOEL.MAK" FOR OUTPUT AS #2
IF NOT EOF(1) THEN INPUT #1, N1$
WHILE NOT EOF(1)
  INPUT #1, N2$
  PRINT #2, N1$
  PRINT N1$
  N1$ = N2$
WEND
CLOSE #1
CLOSE #2
SYSTEM
```

Daarnaast hebt u het batchbestand MAKEBAT.BAT nodig. Dat bestand maakt van de opdrachten in de buffer van DOSKEY een bronbestand, start QBASIC en kopieert het doelbestand uiteindelijk naar een batchbestand:

```
@ECHO OFF
REM MAKEBAT.BAT maakt een batchbestand met behulp van /HISTORY
DOSKEY /HISTORY >C:\TEMP\BRON.MAK
QBASIC /RUN C:\TEMP\VERWRGL.BAS
COPY DOEL.MAK %1.BAT
ECHO "%1.BAT%" aangemaakt
```

Deze wat omslachtige benadering is nodig, omdat QBASIC geen variabelen toestaat in de naam van draaiende programma's. U kunt dus niet rechtstreeks de naam van het batchbestand opgeven. VERWRGL.BAS gebruikt altijd (!) de namen BRON.MAK en DOEL.MAK. Het batchbestand zorgt dan voor de opdracht die het bestand hernoemt.

Tip: Let goed op een juiste opgave van het zoekpad bij beide bestanden. Gebruikt u C:\TEMP, dan dient die directory ook te bestaan. U zou de beide bestanden ook kunnen opslaan in de directory C:\BAT, maar omdat het om tijdelijke bestanden gaat, verdient dat niet de voorkeur.

6 Bestanden en directories overzichtelijk weergeven

De opdracht DIR is ongetwijfeld een van de meest gebruikte opdrachten in MS-DOS. Toch maken gebruikers die onder versie 5.0 werken, vaak niet optimaal gebruik van de mogelijkheden van de opdracht. In dit hoofdstuk zullen we de opdracht uitvoerig bespreken en u een aantal tips aan de hand doen waarmee u meer uit de opdracht kunt halen.

6.1 Een DIR voor iedere situatie

De opdracht DIR zonder parameters is vaak aardig te gebruiken voor diskettes. Voor een harde schijf met enige tientallen megabytes aan gegevens levert DIR op die manier weinig op. Wilt u zien hoeveel bestanden dat wel niet zijn, geef dan eens de volgende opdracht:

```
DIR /S
```

Wacht u rustig af tot DOS alle (!) bestanden van uw harde schijf over uw scherm heeft laten flitsen. Weinig informatief, maar gelukkig kan het ook anders met behulp van opties.

Uitwerking

Bij de bespreking van de onderstaande opties gaan we ervan uit dat C:\ de actuele directory is.

DIR A:\ geeft een overzicht van de hoofddirectory van drive A:

DIR A: geeft een overzicht van de actuele directory van drive A:

DIR TEKSTEN
geeft een overzicht van de subdirectory TEKSTEN, mits die een rechtstreekse subdirectory is van C:\

DIR *.EXE geeft een overzicht van alle bestanden in de actuele directory met de extensie .EXE

DIR F*.* geeft een overzicht van alle bestanden in de actuele directory die beginnen met de letter F

DIR \ geeft een overzicht van de hoofddirectory van de actuele drive

DIR .. geeft een overzicht van de directory waar de actuele directory een subdirectory van is; in dit geval werkt de opdracht niet, omdat we zijn uitgegaan van de hoofddirectory als actuele directory

DIR /P geeft een overzicht van de inhoud van de actuele directory per pagina; een toetsaanslag brengt de volgende pagina op het scherm

DIR /W geeft een overzicht van de inhoud van de actuele directory in vijf kolommen: alleen bestandsnaam en extensie worden weergegeven

DIR /S geeft een overzicht van alle bestanden in de actuele directory en alle bijbehorende subdirectories

DIR /B geeft een overzicht van de inhoud van de actuele directory: alleen bestandsnaam en extensie worden weergegeven

U kunt opties combineren. Zo kunt u bijvoorbeeld met de volgende opdracht alle tekstbestanden op uw harde schijf weergeven, ongeacht de directory waarin ze staan. Overigens wordt wel steeds aangegeven welke directory het systeem leest. Uiteraard krijgt u alleen dan alle tekstbestanden te zien, als u de opdracht vanuit de hoofddirectory invoert.

```
DIR *.TXT /S
```

6.2 DIR optimaal gebruiken

Er zijn nog meer mogelijkheden om de opdracht DIR zo aan te passen dat u er maximaal profijt van kunt trekken, bijvoorbeeld om uit een groep bestanden dat bestand te zoeken dat u het laatst hebt aangemaakt, of het bestand waarvan u nog geen reservekopie op diskette hebt gemaakt.

Uitwerking

Naast de hierboven besproken opties kent MS-DOS 5.0 er nog een aantal waarmee u bestanden gesorteerd kunt weergeven of waarmee u bijvoorbeeld alleen bestanden

weergeeft met bepaalde bestandsattributen.

Hieronder volgt een opsomming van de parameters waarmee u bestanden kunt sorteren bij weergave. U mag desgewenst parameters combineren.

/O:N De bestanden worden gesorteerd op bestandsnaam.

/O:-N De bestanden worden gesorteerd op bestandsnaam in omgekeerde alfabetische volgorde.

/O:E De bestanden worden gesorteerd op extensie.

/O:-E De bestanden worden gesorteerd op extensie in omgekeerde alfabetische volgorde.

/O:D De bestanden worden gesorteerd op datum en tijd.

/O:-D De bestanden worden gesorteerd op datum en tijd in omgekeerde volgorde.

/O:S De bestanden worden gesorteerd op omvang.

/O:-S De bestanden worden gesorteerd op omvang in omgekeerde volgorde.

/O:G De directories worden gegroepeerd weergegeven vóór de bestanden.

/O:-G De directories worden gegroepeerd weergegeven nà de bestanden.

U mag de dubbele punt weglaten.

Wilt u bijvoorbeeld de bestanden groeperen op extensie in alfabetische volgorde en vervolgens ook nog op datum, dan geeft u de volgende opdracht:

```
DIR /OED
```

De volgende rij parameters is bedoeld voor selectie van bestanden op grond van bestandsattributen. Ook nu kunt u weer parameters combineren.

/A:H Alleen bestanden met het attribuut Hidden worden weergegeven.

/A:-H Alleen bestanden waarvan het attribuut Hidden niet is gezet, worden weergegeven.

/A:S Alleen bestanden met het attribuut System worden weergegeven.

/A:-S Alleen bestanden waarvan het attribuut System niet is gezet, worden weergegeven.

/A:A Alleen bestanden met het attribuut Archive worden weergegeven.

/A:-A Alleen bestanden waarvan het attribuut Archive niet is gezet, worden weergegeven.

/A:R Alleen bestanden met het attribuut Read only worden weergegeven.

/A:-R Alleen bestanden waarvan het attribuut Read only niet is gezet, worden weergegeven.

/A:D Alleen namen van directories worden weergegeven.

/A:-D Namen van directories worden niet weergegeven.

Ook nu kunt u de dubbele punt weer weglaten.

Wilt u dus alleen bestanden op het scherm waarvan nog geen reservekopie is gemaakt, dan luidt de opdracht:

```
DIR /A-D-A
```

U kunt deze parameters ook combineren met de eerder besproken parameters. De volgende opdracht geeft alle bestanden met de extensie .BAT waarvan nog geen reservekopie is gemaakt, per pagina weer, gesorteerd op bestandsnaam:

```
DIR /*.BAT /ON /A-A-D /P
```

De bovenstaande opdracht is er zo een die tegenstanders van MS-DOS doet gruwen. Er is echter niets op tegen een dergelijke opdracht op te slaan in een klein batchbestand. Hieronder bespreken we een paar handige batchbestanden voor het gebruik van DIR met verschillende parameters.

Tekstbestanden gesorteerd weergeven

Voor een weergave van alle tekstbestanden op een harde schijf, gesorteerd op naam en datum, gebruikt u het batchbestand DIRTEKST.BAT:

```
@REM ****************************************************
@REM * Dit batchbestand geeft alle tekstbestanden weer  *
@REM *    per pagina, gesorteerd op naam en datum       *
@REM ****************************************************
@DIR *.TXT /OND /A-D /P
```

Alleen bestanden alfabetisch gesorteerd weergeven

Het volgende batchbestand, DIRA.BAT, geeft alleen de bestanden weer in de actuele directory, alfabetisch gesorteerd op naam:

```
@REM *************************************************
@REM *   Dit batchbestand geeft de bestanden weer    *
@REM *  in de actuele directory, gesorteerd op naam  *
@REM *************************************************
@DIR /ON /A-D /P
```

Bestanden sorteren op omvang

Het batchbestand DIRG.BAT sorteert bestanden op omvang:

```
@REM ************************************************
@REM * Dit batchbestand geeft de bestanden weer in  *
@REM *  de actuele directory, gesorteerd op omvang  *
@REM ************************************************
@DIR /OS /A-D /P
```

Bestanden sorteren op datum

Het batchbestand DIRD.BAT sorteert bestanden op datum:

```
@REM ************************************************
@REM * Dit batchbestand geeft de bestanden weer in  *
@REM * de actuele directory, gesorteerd op datum    *
@REM ************************************************
@DIR /OD /A-D /P
```

Directories alfabetisch gesorteerd weergeven

Het batchbestand DIRDIRN.bat geeft directories weer, gesorteerd op naam. De beide directories '.' en '..' worden ook weergegeven, omdat ze voor respectievelijk de actuele en de erboven geplaatste directory staan.

```
@REM **********************************************
@REM * Dit batchbestand geeft alleen directories   *
@REM * weer, gesorteerd op naam                     *
@REM **********************************************
@DIR /ON /AD /P
```

Batchbestanden voor DIR met variabelen

Wilt u bij het gebruik van de hierboven beschreven batchbestanden de mogelijkheid open houden om incidenteel nog meer parameters toe te voegen, dan kunt u in het batchbestand plaatshouders (%1,%2, enzovoort) toevoegen. Het batchbestand dat bestandsnamen alfabetisch gesorteerd weergeeft, ziet er dan als volgt uit:

```
@REM **********************************************
@REM *  Dit batchbestand geeft de bestanden weer    *
@REM * in de actuele directory, gesorteerd op naam  *
@REM **********************************************
@DIR /ON /A-D /P %1 %2 %3
```

Roept u nu het batchbestand op met de opdracht:

```
DIR DIRA *.TXT
```

dan geeft het batchbestand alleen de tekstbestanden alfabetisch gesorteerd weer.

Bestanden sorteren met een DOSKEY-macro

In plaats van een batchbestand kunt u ook een macro aanmaken met DOSKEY. De macro DIRD, die de bestanden steeds gesorteerd op datum weergeeft, ziet er als volgt uit:

```
DOSKEY DIRD=DIR /OD
```

Vanzelfsprekend kunt u alle door ons hierboven besproken batchbestanden vervan-

gen door macro's. U zult ze dan echter bij iedere systeemstart opnieuw moeten invoeren, tenzij u ze opneemt in AUTOEXEC.BAT of een batchbestand MACRO.BAT. Of u met batchbestanden dan wel met macro's werkt, is afhankelijk van de geheugenruimte die u ter beschikking hebt: DOSKEY gebruikt vier Kb van het geheugen en iedere macro die u aanmaakt, gebruikt eveneens geheugen. Als u dus veel macro's opneemt in AUTOEXEC.BAT, kan het zijn dat u problemen krijgt met applicaties.

6.3 Parameters vastleggen in de systeemvariabele DIRCMD

Een directory bevat al gauw meer dan zo'n 25 items. De optie /P, waarmee u een overzicht per pagina op het scherm brengt, zou dus eigenlijk beter standaard aan de opdracht DIR kunnen worden gekoppeld. MS-DOS 5.0 kent een systeemvariabele DIRCMD waarmee u dat kunt doen.

Uitwerking

Om een overzicht van een directory per pagina op het scherm weer te geven, hebt u de volgende opdracht nodig:

```
SET DIRCMD=/P
```

Nadat u de opdracht hebt ingevoerd, gedraagt de opdracht DIR zich alsof u DIR /P hebt getypt.

Toelichting

De systeemvariabele DIRCMD werkt net als de andere systeemvariabelen (PATH, COMSPEC, enzovoort). Als u de variabele hebt geïnstalleerd, past het besturingssysteem zich aan. U kunt dus via de variabele opgeven dat het besturingssysteem bij verstek bestanden gesorteerd op naam en per pagina moet weergeven, als u de variabele als volgt definieert:

```
SET DIRCMD=/ON /P
```

Een tweede stap is het opnemen van de opdracht in het bestand AUTOEXEC.BAT. Het systeem zal nu iedere keer dat u het start, de variabele initialiseren en de opdracht DIR overeenkomstig uitvoeren.

U mag, ook nadat u de opdracht DIR via de systeemvariabele hebt aangepast, nog steeds opties opgeven bij DIR:

```
DIR *.TXT
```

Het bovenstaande geeft alleen de tekstbestanden weer. Onder invloed van de systeemvariabele gebeurt dat echter alfabetisch gesorteerd.

Zit de systeemvariabele u op een gegeven moment in de weg, dan kunt u hem eenvoudig uitschakelen met de opdracht:

```
DIRCMD=
```

Gaat het er u echter om alleen de weergave per pagina te onderdrukken, dan mag u ook de optie -P toevoegen, waarmee u op dat moment de in de systeemvariabele opgenomen parameter /P onderdrukt.

DIRCMD in een batchbestand

Als u DIRCMD opneemt in het bestand AUTOEXEC.BAT en u hebt de hierboven beschreven opdracht 'DIRCMD=' uitgevoerd, dan zult u het bestand AUTOEXEC.BAT opnieuw moeten activeren om DIRCMD weer te activeren. Dat is lastig. Handiger is het wellicht om twee batchbestanden aan te maken: DIRCMDON.BAT en DIRCMDOF.bat. Daarmee schakelt u de systeemvariabele naar behoefte in en uit.

DIRCMDON.BAT

```
@REM *******************************************
@REM * Weergave van directories onder invloed  *
@REM *     van de systeemvariabele DIRCMD      *
@REM *******************************************
@DIRCMD=/ON /P
@DIR
```

DIRCMDOF.BAT

```
@REM *********************************************
@REM * Uitschakelen van de systeemvariabele DIRCMD  *
```

```
@REM ************************************************
@DIRCMD=
@DIR
```

Toelichting

In het eerste batchbestand wordt de systeemvariabele DIRCMD geïnitialiseerd. Onmiddellijk daarna wordt een overzicht van de inhoud van de actuele directory weergegeven, die laat zien dat de variabele actief is. Vanaf dat moment blijft de variabele actief totdat u het tweede batchbestand oproept. Daarin wordt de variabele gewist en om te laten zien dat dat is gebeurd, volgt opnieuw een overzicht van de inhoud van de actuele directory.

6.4 De omvang van een groep bestanden uitzoeken

Bij het kopiëren van bestanden van de harde schijf naar een diskette, is het erg vervelend om halverwege te worden geconfronteerd met de melding dat de diskette vol is. Vergelijkbaar daarmee is de situatie dat u bij het maken van een backup van de harde schijf van tevoren niet weet hoeveel diskettes u nodig hebt. Een goedgekozen DIR-opdracht helpt u verder.

Uitwerking

Als u in MS-DOS 5.0 de opdracht DIR geeft, verschijnt onder het overzicht nog een boodschap die er bijvoorbeeld als volgt uitziet:

```
33 file(s)      88499 bytes
              3131392 bytes free
```

Op de eerste regel wordt meegedeeld hoeveel bestanden de directory bevat; daarachter staat hoeveel bytes die bestanden samen beslaan. Op de tweede regel staat hoeveel ruimte er op de (harde) schijf nog vrij is.

Met behulp van deze gegevens kunt u uitrekenen of u een groep bestanden nog naar een schijf kunt kopiëren. Stel dat u de volgende opdracht hebt gegeven:

```
DIR /S
```

U kunt dan uitrekenen hoeveel diskettes u nodig hebt voor het maken van een backup.

Wilt u bijvoorbeeld alle tekstbestanden uit een bepaalde directory kopiëren naar een lege diskette, maar u wilt eerst controleren of ze er wel op passen, dan activeert u die directory en geeft u de volgende opdracht:

```
DIR *.TXT
```

Op de regel met het aantal bestanden staat nu hoeveel bytes ze in beslag nemen.

Tip: DIR vertelt u hoe groot een bestand is, maar zegt er niet bij hoeveel ruimte zo'n bestand op een schijf in beslag neemt. Dat laatste is namelijk afhankelijk van het aantal bytes dat MS-DOS voor het opslaan van een bestand gebruikt en kan variëren van 512 tot 4096 bytes.

U kunt dit gemakkelijk testen door het bestand AUTOEXEC.BAT te kopiëren naar een lege diskette. De opdracht DIR zal aangeven dat het bestand zo'n 50 tot 500 bytes groot is. Test u nu de schijf met de opdracht CHKDSK, dan krijgt u onder andere bijvoorbeeld de melding:

```
1024 bytes in 1 userfiles
```

Om werkelijk zeker te weten dat alle bestanden op een diskette passen dient u dus eerst de grootte van alle bestanden af te ronden op een veelvoud van 512, 1024, 2048 of 4096 bytes. In de praktijk werkt het erg goed om een marge van een paar Kb aan te houden.

Misschien stoort het u dat u bij dit werk ook nog alle bestanden te zien krijgt. Gebruik dan het volgende batchbestand:

```
@REM ******************************
@REM *  Berekening van de omvang  *
@REM *  van een groep bestanden   *
@REM ******************************
@ECHO OFF
IF "%1"=="" GOTO Fout
DIR %1 %2 | FIND "file(s)"
GOTO Einde
:Fout
ECHO Berekening van de omvang van een groep bestanden
ECHO Syntaxis: DIR BESTAND.EXT (wildcards toegestaan)
:Einde
@ECHO ON
```

Let op Het kan zijn dat de werking van het batchbestand wordt verstoord door de instelling van de systeemvariabele DIRCMD (zie paragraaf 6.3). Schakel die variabele dan tijdelijk uit om het batchbestand te testen.

Toelichting

Dit batchbestand biedt u de mogelijkheid bestanden uit te kiezen. Wilt u alle bestanden van een bepaalde directory, dan geeft u het zoekpad op en gebruikt u voor de bestandsnaam de wildcards '*.*'.

Hebt u het batchbestand laten berekenen hoeveel bytes een groep bestanden in beslag neemt, dan kunt u vervolgens kijken of er op de diskette waar u de bestanden naar toe wilt kopiëren, genoeg ruimte is. U geeft weer de opdracht DIR, nu voor de drive met de diskette, en op de onderste regel staat de vrije ruimte op de diskette vermeld.

Wilt u alle bestanden van de harde schijf, dan activeert u de hoofddirectory en voegt u de optie '/S' toe. In dat geval krijgt u een lijst te zien, waarin het aantal bytes per directory wordt aangegeven. De laatste regel bevat het totaal. Daaruit kunt u afleiden hoeveel diskettes u nodig hebt door dit totaal te delen door de capaciteit van de diskettes.

6.5 Subdirectories zoeken

Bij een redelijk gevulde harde schijf kan maar zo een subdirectory wegraken. In de meeste gevallen is er niets ernstigs aan de hand: de subdirectory bevindt zich domweg in een andere directory dan u dacht. Met behulp van de opdracht DIR spoort u hem snel weer op.

Uitwerking

De opdracht DIR kan u bij het zoeken naar een directory helpen, omdat u met de optie /AD bestanden kunt uitsluiten en de naam van de directory die u zoekt, mag opgeven als parameter. Daarbij hoort het volgende batchbestand WAARDIR.BAT:

```
@REM ******************************************
@REM * Het zoekpad naar een directory bepalen  *
@REM ******************************************
@ECHO OFF
```

```
IF "%1"=="" GOTO Fout
CD \
DIR /S /AD | FIND "%1" /I | MORE
GOTO Einde
:Fout
ECHO Het batchbestand zoekt een directory met subdirectories
ECHO Syntaxis: WAARDIR DIRECTORY (geen wildcards toegestaan)
:Einde
@ECHO ON
```

Als u de bestanden van PC TOOLS hebt opgeslagen in een directory PCTOOLS die een subdirectory is van C:\TOOLS, kunt u de volgende opdracht invoeren:

```
WAARDIR PCTOOLS
```

Na enig zoekwerk verschijnt dan de volgende melding op uw scherm:

```
Directory of C:\TOOLS\PCTOOLS
```

Weet u niet meer precies hoe de directory heet, dan kunt u volstaan met de eerste letters ervan. Zijn er meer directories die met die letters beginnen, dan somt het batchbestand die directories op. Een opgave van 'TE' zou bijvoorbeeld het volgende resultaat kunnen hebben:

```
Directory of C:\TEKSTEN
Directory of C:\TEMP
```

Toelichting

Het batchbestand activeert eerst de hoofddirectory van de actuele drive. Vervolgens maakt het met de opdracht DIR /S een overzicht van alle subdirectories. De optie /AD zorgt ervoor dat bestanden worden uitgesloten. De opdracht FIND doorzoekt daarna de uitvoer naar een bepaalde variabele, in dit geval 'PCTOOLS'. De optie /I zorgt er daarbij voor dat er geen onderscheid wordt gemaakt tussen hoofdletters en kleine letters.

6.6 Alle bestandsnamen van de harde schijf alfabetisch sorteren

Hiervoor hebben we opties besproken waarmee u een overzicht van de inhoud van een directory kunt maken, alfabetisch gesorteerd op naam. In deze paragraaf gaan we nog wat dieper in op die materie: we maken een gesorteerd overzicht van alle bestanden en filteren daar alle niet ter zake doende informatie uit.

Uitwerking

We laten u twee mogelijkheden zien om alle bestandsnamen van de harde schijf alfabetisch te sorteren: met behulp van een batchbestand en via de DOS Shell.

Maak een batchbestand ALLEBST.BAT met de volgende opdracht:

```
DIR /A-D /S | FIND /V "e" | FIND "  " | SORT | MORE
```

Wilt u het resultaat liever rechtstreeks naar een bestand schrijven, dan luidt de opdracht als volgt:

```
DIR /A-D /S | FIND /V "e" | FIND "  " | SORT > C:\TEMP\ALLEBST.TXT
```

Toelichting

In de eerste plaats geeft u met de opdracht DIR /A-D /S aan dat u een overzicht wilt van alle bestanden uit alle directories. Het is van belang dat u voor die tijd de hoofddirectory activeert, want de uiteindelijke uitvoer laat op geen enkele wijze zien welke directories zijn doorzocht. Gebruikt u een batchbestand, dan plaatst u op de regel voor de DIR-opdracht nog een opdracht:

```
CD \
```

De uitvoer van de opdracht DIR wordt vervolgens omgeleid. We maken gebruik van de eigenschap van MS-DOS om onderscheid te maken tussen hoofdletters en kleine letters. Met de opdracht FIND zoeken we alle regels in de uitvoer waarin de kleine letter 'e' voorkomt. De optie /V zorgt ervoor dat die regels uit de uitvoer worden gefilterd. Omdat alle meldingen die de opdracht DIR ten gevolge heeft, de kleine letter 'e' gebruiken, houden we dus bijna een schone lijst over. De opdracht DIR plaatst echter in een overzicht ook altijd een paar lege regels. Die kunnen we verwijderen

door de te zoeken naar regels met twee opeenvolgende spaties. Wat dan overblijft, vormt de invoer voor de opdracht SORT, die alle bestandsnamen alfabetisch sorteert.

De DOS Shell kent voor dit werk een aparte functie. Open het menu View en kies All files. Het scherm verandert: naast elkaar worden twee vensters geopend. In het rechter venster verschijnt een lijst met alle bestanden van de harde schijf, gerangschikt in alfabetische volgorde. Het linker venster verschaft informatie over het bestand dat in het rechter venster is gemarkeerd. Die informatie betreft onder andere de bestandsattributen, de grootte van het bestand en het zoekpad naar het bestand. Dat laatste geeft u tegelijkertijd een alternatief voor het batchbestand dat we in de volgende paragraaf zullen bespreken.

6.7 Alle directories doorzoeken naar een bestand

In paragraaf 6.5 hebben we u een methode aan de hand gedaan om snel een directory te zoeken. Verloren geraakte bestanden vormen een nog vervelender probleem. Ook die zijn vaak niet echt weg: ze bevinden zich domweg niet in de door u veronderstelde directory. Hieronder vindt u een methode om snel bestanden op te sporen.

Uitwerking

Het batchbestand WAARBST.BAT doorzoekt alle directories van de actuele drive naar een opgegeven bestandsnaam. In plaats van de hele naam mag u ook het eerste deel ervan opgeven.

```
@REM ****************************************************************
@REM * Het batchbestand doorzoekt alle directories naar een bestand*
@REM ****************************************************************
@ECHO OFF
IF "%1"=="" GOTO Fout
CD \
DIR \%1 /S /P
GOTO Einde
:Fout
ECHO Het batchbestand doorzoekt alle directories naar een bestand
ECHO Syntaxis: WAARBST BESTAND.EXT (alleen beginletters toegestaan)
:Einde
@ECHO ON
```

Het bestand BOEK.TXT zoekt u als volgt op:

```
WAARBST BOEK.TXT
```

Wilt u alle bestanden die iets met boeken te maken hebben, dan kunt u de volgende opdracht geven:

```
WAARBST BO*.*
```

Alle tekstbestanden zoekt u met:

```
WAARBST *.TXT
```

6.8 Naar een string zoeken in bestanden

U hebt een tekst geschreven en opgeslagen op de harde schijf. Nu bent u de bestandsnaam vergeten en weet u ook niet meer in welke directory het bestand staat. Gelukkig gaat de tekst over ‘Pseudo- ecclectische Kruisspin-variaties’, wat dan ook als titel boven de tekst staat. Er is een batchbestand waarmee u die tekst terug kunt vinden.

Uitwerking

Het onderstaande batchbestand doorzoekt directories naar een *string*. Een string is een reeks tekens.

```
@REM ***************************************************************
@REM * Het batchbestand zoekt in een selectie van bestanden een    *
@REM * string                                                      *
@REM ****************************************************************
@ECHO OFF
IF "%1"=="" GOTO Fout
IF "%2"=="" GOTO Fout
FOR %%a IN (%1) DO FIND "%2" %%a
GOTO Einde
:Fout
ECHO Het batchbestand zoekt in een selectie van bestanden een string
ECHO Syntaxis: FINDSTR BESTAND.EXT STRING (wildcards toegestaan)
:Einde
@ECHO ON
```

Geef bij de opdracht om het batchbestand te starten twee parameters op: de eerste selecteert de bestanden waarin moet worden gezocht; de tweede is de zoekstring. Voor de bestanden mag u een zoekpad opgeven. Plaats de zoekstring niet tussen aanhalingstekens, maar gebruik wel hoofdletters en kleine letters op de juiste plaats. U vindt de tekst over kruisspinnen ongetwijfeld met een van de volgende opdrachten:

```
FINDSTR *.TXT Pseudo
FINDSTR *.TXT ecclectische
FINDSTR *.TXT Kruisspin
FINDSTR *.TXT -variaties
```

Weet u zeker dat het bestand in een directory C:\TEKSTEN moet staan, dan kunt u het zoekpad opgeven; misschien is de extensie wel .DOC, dan geeft u ook de extensie op:

```
FINDSTR C:\TEKSTEN\*.DOC Kruisspin
```

Helaas is het niet zonder meer mogelijk de weergave op het scherm aan te houden met PAUSE of MORE. MS-DOS staat die opdrachten niet toe na de opdracht FIND. Als u dat stoort, kunt u de volgende aangepaste versie van FINDSTR.BAT gebruiken.

```
@REM ****************************************************************
@REM * Het batchbestand zoekt in een selectie van bestanden een     *
@REM * string                                                       *
@REM ****************************************************************
@ECHO OFF
IF "%1"=="" GOTO Fout
IF "%2"=="" GOTO Fout
ECHO >FIND.TXT---------------het zoekresultaat-------------
FOR %%a IN (%1) DO FIND >>FIND.TXT "%2" %%a
GOTO Einde
:Fout
ECHO Het batchbestand zoekt in een selectie van bestanden een string
ECHO Syntaxis: FINDSTR BESTAND.EXT STRING (wildcards toegestaan)
:Einde
TYPE FIND.TXT | MORE
@ECHO ON
```

Toelichting

Het batchbestand controleert eerst of beide parameters zijn opgegeven. Daarna worden met de FOR-opdracht alle bestanden geselecteerd die moeten worden doorzocht. Die worden dan samen met de zoekstring doorgegeven aan de opdracht FIND. Op zijn beurt doorzoekt die de bestanden en voert steeds de regel waarin de zoekstring is gevonden, uit naar het scherm.

Gaat het om een string die in veel bestanden voorkomt, dan scrollt de uitvoer over uw scherm, zonder dat u daar iets aan kunt doen. Het tweede batchbestand schrijft daarom de uitvoer naar een bestand FIND.TXT. De eerste regel van dit bestand luidt 'het zoekresultaat'. Iedere nieuwe vondst wordt met '>>FIND.TXT' aan het einde van dit bestand toegevoegd. Aan het einde van het batchbestand wordt FIND.TXT per pagina op het scherm gezet. Wilt u een string zoeken in alle bestanden van alle directories van de harde schijf, dan kunt u daarvoor gebruik maken van een tip die we beschrijven in paragraaf 8.3.

7 Optimaal geheugenbeheer

Dat het ene geheugen het andere niet is, hebt u ongetwijfeld al begrepen. Veel PC's met een geheugen van twee Mb houden daarvan maar zo'n 500 Kb over als u de machine hebt gestart. In dit hoofdstuk zullen we tips en trucs laten zien die met geheugenbeheer te maken hebben. Daarmee houdt u hopelijk uiteindelijk meer beschikbaar geheugen over.

We kunnen hier onmogelijk op alle fundamentele begrippen ingaan. Voelt u zich onzeker als termen als extended memory, XMS-geheugen, upper memory en high memory ter sprake komen, dan zult u eigenlijk toch een boek moeten raadplegen dat zich wat uitvoeriger met geheugenproblematiek bezig houdt. We raden u aan bijvoorbeeld hoofdstuk 12 van 'Het grote DOS 5.0-boek' (A.W. Bruna 1991) te lezen.

Tip: Maak voordat u verder gaat, een systeemdiskette voor MS-DOS 5.0, dat wil zeggen een diskette met de bestanden COMMAND.COM (en de bijbehorende systeembestanden), CONFIG.SYS en AUTOEXEC.BAT. Zie paragraaf 7.6 voor meer bijzonderheden over het maken van zo'n diskette. U hebt die diskette nodig als de PC blijft hangen. Vooral als dat het gevolg is van een wijziging in het bestand CONFIG.SYS, kan het gebeuren dat u het systeem vervolgens op geen enkele manier meer kunt starten vanaf de harde schijf!

7.1 Geheugenruimte besparen

Moderne software vraagt om veel geheugen. Vrijwel alle PC's hebben vandaag de dag minstens 640 Kb en toch gebeurt het regelmatig dat een applicatie meldt dat er niet genoeg vrij geheugen is. U hoeft dan niet meteen over te gaan tot de aanschaf van een PC met 4 Mb geheugen; er zijn veel manieren om het beschikbare geheugen doeltreffender te gebruiken.

Uitwerking

Veel van wat u kunt doen om meer geheugen beschikbaar te maken hangt af van het type PC dat u gebruikt. Maar los daarvan kunt u proberen het geheugen zo in te delen dat er geen loze ruimten ontstaan. Met dat laatste zullen we ons in de komende paragrafen bezighouden.

Het eerste wat u kunt doen, is kritisch kijken naar de programma's die u laadt: maakt u een groot bestand AUTOEXEC.BAT of CONFIG.SYS, dan vreten die geheugen. Daarnaast legt elk TSR-programma dat u voor alle zekerheid installeert, beslag op een deel van dat kostbare geheugen.

Op een computer zonder geheugenuitbreiding zonder aanpassingen aan de systeembestanden kunt u rekenen op 578 Kb werkgeheugen (592 848 bytes). Afhankelijk van bijvoorbeeld het aantal drives waarover de computer beschikt, kan dat iets minder zijn. Veel meer zal het nooit zijn.

In het algemeen valt niet te zeggen welke programma's u vooral wel of niet moet installeren, maar er zijn toch wel een aantal opdrachten die u kunt gebruiken zonder dat ze een beroep op het geheugen doen. De onderstaande opdrachten zijn interne DOS-opdrachten die deel uitmaken van de systeembestanden.

opdracht	**omschrijving**
BREAK	uitvoering van opdrachten onderbreken
COMSPEC	systeemvariabele voor het besturingssysteem
COUNTRY	instelling van taalgebonden tekens en notaties
ECHO	uitvoer van meldingen naar het scherm
PATH	definitie van het zoekpad
PROMPT	definitie van het teken van de systeemprompt
REM	uitvoer van een commentaarregel
SET	definitie van een systeemvariabele
VER	opgave van de versie van het besturingssysteem
VERIFY	controle bij schrijfhandelingen naar schijf
SHELL	een besturingssysteem installeren

Weliswaar kan het u een paar byte geheugenruimte kosten als u een variabele als COMSPEC of PROMPT definieert, maar dat heeft geen praktische betekenis. Installeert u KEYB omdat u bijzondere tekens wilt typen, dan kost u ongeveer 6 Kb en dat is wel wat meer.

Hieronder behandelen we de belangrijkste opdrachten en instellingen voor de bestanden AUTOEXEC.BAT en CONFIG.SYS, waarbij we steeds aangeven hoeveel geheugen daarvoor nodig is. Daarna bespreken we kort minder vaak gebruikte DOS-opdrachten die mogelijkerwijs deel uitmaken van uw systeembestanden. U bent dan zelf in staat in te schatten wat u werkelijk wilt laden en hoeveel geheugenruimte u daarvoor over hebt ten koste van applicaties.

Tip: Aan het eind van deze paragraaf vatten we alle informatie samen in een tabel.

APPENDzoekpad voor gegevensbestanden, de tegenhanger van PATH, waarmee u het zoekpad voor programmabestanden vastlegt. APPEND gebruikt 9280 bytes geheugenruimte; een deel van het programma blijft resident in het geheugen aanwezig.

BUFFERS Het aantal buffers dat wordt gebruikt bij lees- en schrijfhandelingen. Iedere buffer reserveert 532 bytes geheugenruimte en kan 512 bytes aan gegevens opslaan. 20 bytes worden dus gebruikt voor de besturing. MS-DOS houdt bij verstek 15 buffers aan in een systeem met meer dan 511 Kb werkgeheugen. In het totaal is dat 7980 bytes. Niets weerhoudt u ervan slechts één buffer te gebruiken (in het bestand CONFIG.SYS: BUFFERS = 1) om geheugenruimte te besparen; de snelheid van uw computer loopt dan echter wel drastisch terug.

EMM386.SYS Device driver voor het beheer van expanded memory en upper memory. Het verlies van 8896 bytes in het werkgeheugen wordt ruimschoots goed gemaakt door de winst aan geheugenruimte boven de grens van 640 Kb. De driver werkt alleen op computers met een 80386-processor, die zijn uitgerust met extra geheugen.

FASTOPEN FASTOPEN versnelt het opnieuw openen van bestanden. Een deel van het programma blijft resident in het geheugen en neemt daar minstens 3680 bytes in beslag. FASTOPEN houdt bepaalde gegevens van programma's bij en heeft daartoe voor iedere bestand 48 bytes nodig.

FCBS Een verouderd besturingsmechanisme voor geopende bestanden, dat vandaag de dag nauwelijks nog wordt gebruikt. Bij verstek hanteert MS-DOS vier FCBS (File Control BlockS), die elk 64 bytes nodig hebben. Elke volgende FCBS heeft eveneens 64 bytes nodig.

FILES Het maximale aantal bestanden dat tegelijkertijd open mag zijn. Voor het beheer van ieder geopend bestand zijn 64 bytes nodig. De kleinste geldige waarde voor FILES is acht. Als de waarde van FILES te klein is, meldt MS-DOS dat. In de meeste gevallen zult u dan eerst de waarde van FILES moeten verhogen en het systeem opnieuw starten voordat u verder kunt.

GRAFTABL	Een programma dat de grafische tekenset laadt, waarmee tekens met een ASCII-code hoger dan 128 ook in de grafische modus kunnen worden weergegeven. GRAFTABL verbruikt 1296 bytes.
GRAPHICS	een grafisch printerprogramma dat 5968 bytes nodig heeft in het geheugen. Met het programma kunt u ook screendumps maken van het scherm in de grafische modus.
HIMEM.SYS	Device driver voor het XMS-geheugen. Het neemt 1148 bytes in beslag, maar is voor het gebruik van Windows (en veel andere moderne applicaties) onontbeerlijk. De driver werkt alleen op een PC met een geheugenuitbreiding.
KEYB	De instelling van het toetsenbord, vooral nodig voor bijvoorbeeld Duitstalige landen. KEYB verbruikt 6320 bytes.
LASTDRIVE	Het maximale aantal drives dat het systeem ondersteunt. Iedere drive kost 80 bytes. Bij verstek is LASTDRIVE ingesteld op vijf drives. In de meeste gevallen is dat toereikend. Verandert u de waarde: LASTDRIVE=F, dan kost u dat dus 80 bytes.
PRINT	PRINT installeert een *queue* voor de printer. U kunt daarin een aantal printopdrachten plaatsen, die achter elkaar worden uitgevoerd, terwijl u verder werkt. Het residente deel van PRINT verbruikt 5808 bytes.
RAMDRIVE.SYS	Een resident programma dat een RAM-disk installeert, dat wil zeggen een deel van het geheugen reserveert en beschouwt als drive met een eigen drive-letter. Het programma heeft daarvoor in het werkgeheugen 3424 bytes nodig. Installeert u een RAM-disk in het extended memory dan dient ook HIMEM.SYS te zijn geïnstalleerd. De RAM-disk is minstens 64 Kb groot. Installatie in het werkgeheugen heeft dan ook meestal geen zin vanwege het verlies aan geheugenruimte.
SETVER	SETVER maakt het mogelijk aan programma's die dat nodig hebben, een ander versienummer door te geven van MS-DOS. In dat geval moet SETVER als device driver zijn geïnstalleerd in het bestand CONFIG.SYS. Dat kost 400 bytes.
SHARE	Een aantal controle-routines die worden uitgevoerd onder bepaalde omstandigheden in een multi-user omgeving of bij multi-

tasking. Alleen in die gevallen hoeft u SHARE te installeren ten koste van 5880 bytes.

SMARTDRV.SYS Device driver voor een snellere toegang tot bestanden bij lees- en schrijfhandelingen; indien mogelijk wordt gebruik gemaakt van een buffer om te lezen of te schrijven. Een deel van SMARTDRV.SYS blijft resident in het werkgeheugen: 15 552 bytes. Installeren van SMARTDRV.SYS in het geheugen onder 640 Kb heeft geen zin: de buffers hebben minstens 256 Kb nodig, zodat er geen effectief werkgeheugen over blijft. Bij installatie in extended memory moet ook HIMEM.SYS zijn geïnstalleerd.

STACKS Gereserveerd geheugen voor hardware-interrupts. Bij verstek is ruimte gereserveerd voor vier interrupts. In de meeste gevallen kunt u rustig STACKS=0,0 opgeven; u bespaart daar 1872 bytes mee.

De bovenstaande opdrachten en programma's komen veelvuldig voor. Het rijtje hieronder wat minder.

ANSI.SYS Device driver voor het scherm en het toetsenbord. Verbruik: 4208 bytes.

ASSIGN Drive-identificatie wijzigen. Verbruik: 1680 bytes.

COMMAND Opdracht om het besturingssysteem te starten. Een tweede besturingssysteem heeft onder MS-DOS 5.0 maar 3184 bytes nodig, omdat het de meeste taken delegeert aan het eerste besturingssysteem. Dat betekent overigens wel dat beide besturingssystemen identiek moeten zijn.

DEVICE Installatie van device drivers. DEVICE is alleen de opdracht om een device driver te laden. Het geheugenverbruik is afhankelijk van de device driver.

DISPLAY.SYS Device driver voor de ondersteuning van tekensets op het scherm. Verbruik: 4752 bytes.

DRIVER.SYS Device driver voor het identificeren van extra drives. Verbruik: 240 bytes.

DRIVPARM	Opgave van parameters voor drives die geen geheugenruimte verbruikt.
INSTALL	Opdracht om TSR-programma's al in het bestand CONFIG.SYS te installeren, die op zich geen geheugen verbruikt. De benodigde geheugenruimte is afhankelijk van het TSR-programma.
JOIN	Opdracht om een zoekpad aan een drive te koppelen die geen geheugenruimte verbruikt.
MODE	MODE speelt een rol bij de omleiding van een seriële naar een parallelle interface. Een deel van MODE wordt resident geladen en heeft 496 bytes geheugenruimte nodig. Gebruikt u MODE bij de installatie van device drivers, dan heeft de opdracht geen geheugen nodig.
NLSFUNC	Laden van landgebonden gegevens kost 2880 bytes.
SUBST	Een directory identificeren als drive; SUBST verbruikt geen geheugen.

Tip: U kunt, als u alles wat hierboven aan bytes staat bij elkaar optelt, nogal wat geheugenruimte besparen door geen onnodige drivers te installeren en noodzakelijke instellingen zo klein mogelijk te houden.

Overzicht van opdrachten en het geheugenverbruik

We hebben alle besproken opdrachten en programma's in de onderstaande tabel bij elkaar gezet op alfabetische volgorde. U kunt de tabel erbij nemen en de bestanden AUTOEXEC.BAT en CONFIG.SYS eens kritisch bekijken. Als een programma resident in het geheugen wordt geladen, kunt u ervan uitgaan dat het in de tweede kolom vermelde aantal bytes nodig is. Als u een programma op een bepaalde waarde kan worden ingesteld, geeft de derde kolom u een hint voor een algemeen bruikbare instelling. Onder de tabel lichten we een paar bijzonderheden toe.

opdracht	geheugenruimte	instelling
ANSI.SYS	4208	0
APPEND	9280	0
ASSIGN	1680	0
BREAK	0	0
BUFFERS	0	0
COMMAND	3184	1)
COMSPEC	0	0
COUNTRY	0	0
DEVICE	0	0
DISPLAY.SYS	4752	0
DRIVER.SYS	0	240
DRIVPARM	0	0
ECHO	0	0
EMM386.SYS	8896	2)
FASTOPEN	3680	48
FCBS	0	64
FILES	0	64
GRAFTABL	1296	0
GRAPHICS	5968	0
HIMEM.SYS	1184	2)
INSTALL	0	0
JOIN	0	0
KEYB	6320	0

opdracht	geheugenruimte	instelling
LASTDRIVE	0	80
MODE	496	3)
NLSFUNC	1880	0
PATH	0	0
PRINT	5808	0
PROMPT	0	0
RAMDRIVE.SYS	3424	4)
REM	0	0
SETVER	400	0
SET	0	0
SHARE	5880	18
SHELL	0	0
SMARTDRV.SYS	15552	0
STACKS	0	468
SUBST	0	0
VER	0	0
VERIFY	0	0

1) Elke volgende versie delegeert een groot aantal taken aan het eerste besturingssysteem en heeft dus maar weinig geheugen nodig.

2) Kost geheugen, maar stelt een veelvoud aan geheugen beschikbaar.

3) Alleen als MODE resident wordt geladen.

4) Vereist in het werkgeheugen minstens 64 Kb; daarom bij voorkeur installeren in een geheugenuitbreiding.

7.2 De beste configuratie voor een AT met 1 Mb RAM

Veel AT's worden geleverd met één Mb geheugen. Oudere versies van MS-DOS kunnen in principe slechts 640 Kb geheugen adresseren, zodat een gebruiker van zo'n computer al gauw zijn toevlucht zoekt in een RAM-disk en een *Cache-geheugen*. MS-DOS 5.0 kan wel wat meer.

Uitwerking

In de eerste plaats kan de device driver HIMEM.SYS meer geheugen toegankelijk maken voor applicaties. De driver definieert extended memory als zogenaamd XMS-geheugen en maakt daardoor naast *high memory* ook *upper memory* toegankelijk. We zullen wat dieper op die begrippen ingaan. U activeert HIMEM.SYS in het bestand CONFIG.SYS met de opdracht:

```
DEVICE=HIMEM.SYS
```

Dat heeft als resultaat dat HIMEM na het starten van de computer 64 Kb van de geheugenuitbreiding beschikbaar stelt als high memory en de rest als XMS-geheugen.

Alleen programma's die op de juiste manier zijn geprogrammeerd, kunnen gebruik maken van het high memory. Dat is bijvoorbeeld MS-DOS 5.0, maar ook Windows 3.0. Vooral veel oudere applicaties kunnen het high memory niet adresseren.

Om MS-DOS gebruik te laten maken van het high memory is nog een tweede opdracht nodig:

```
DOS=HIGH
```

Het besturingssysteem krijgt daarmee de opdracht een groot deel van zichzelf in het high memory op te slaan. Het normale werkgeheugen wordt zo voor zo'n 40 Kb ontlast, zodat een computer met standaardinstellingen (FILES, BUFFERS, enzovoort) tegen de 600 Kb werkgeheugen vrij houdt voor applicaties.

Het geheugen ziet er nu als volgt uit:

- 640 Kb werkgeheugen, waarvan ongeveer 600 Kb beschikbaar (afhankelijk van de instellingen in de bestanden AUTOEXEC.BAT en CONFIG.SYS).

- 64 Kb high memory (voor de meeste applicaties niet toegankelijk); in ons voorbeeld gebruikt voor het opslaan van de DOS-kern.
- 320 Kb XMS-geheugen.

Met de opdracht MEM kunt u een overzicht van dit geheugen op uw scherm oproepen.

```
655360 bytes total conventional memory
655360 bytes available to MS-DOS
594000 largest executable program size

393216 bytes total contiguous extended memory
     0 bytes available contiguous extended memory
327680 bytes available XMS memory
       MS-DOS resident in High Memory Area
```

Toelichting

De computer beschikt over 640 Kb (655 360 bytes; één Kb is 1024 bytes) werkgeheugen of conventioneel geheugen. In het lijstje hierboven is nog maar 580 Kb (594 000 bytes) beschikbaar voor applicaties, omdat DOS er een deel van gebruikt en omdat onder andere DOSKEY is geïnstalleerd. De totale omvang van de geheugenuitbreiding is 393 216 bytes (384 Kb), oftewel precies het verschil tussen het werkgeheugen en het totale geheugen van 1 Mb (1024 Kb). Van die 384 Kb is 64 Kb gebruikt voor het high memory, zodat voor het XMS-geheugen 320 Kb (327 680 bytes) overblijft. Dit geheugen kunt u gebruiken voor het installeren van een Cachegeheugen of een RAM-disk.

De eerste 64 Kb van het extended memory vanaf 1 Mb kunnen door de processor van een AT gewoon worden geadresseerd, zonder dat de processor daarvoor hoeft over te schakelen naar de *protected mode*. Het voert te ver om hier op de louter technische achtergronden daarvan in te gaan. Laten we volstaan met te zeggen dat het berust op een kleine incompatibiliteit tussen de 80286- en de 8086/8088-processoren, die te maken heeft met de adresleiding A20. Die kan in een computer met een 80286-processor gebruikt worden, waardoor in één klap 64 Kb geheugen beschikbaar komt.

High memory is dus een speciale vorm van extended memory. High memory kan dan ook alleen worden gecreëerd op een systeem met extended memory.

7.3 De beste configuratie voor een 80386 met 1 Mb RAM

U hebt nog meer mogelijkheden op een systeem dat is uitgerust met een van de volgende processoren: 80386, 80386SX, 80386DX, 80486 of 80486SX. Die kunnen namelijk ook gebruik maken van braakliggende geheugengebieden tussen 640 Kb en 1 Mb.

Uitwerking

Het geheugengebied tussen 640 Kb en 1 Mb is eigenlijk gereserveerd voor gebruik door de grafische kaart, het BIOS-ROM en dergelijke. Het heet *upper memory*, omdat het zich in het bovenste (Engels: upper) deel van het geheugen bevindt.

Om het upper memory van een 386 te gebruiken, gaat u als volgt te werk:

- Activeer in de CONFIG.SYS de device driver HIMEM.SYS.
- Activeer daarna de device driver EMM386.EXE, eveneens in de CONFIG.SYS.
- Geef DOS opdracht het upper memory waar mogelijk te gebruiken met DOS=HIGH,UMB of DOS=UMB.

In het bestand CONFIG.SYS kunnen die regels er als volgt uitzien:

```
DEVICE=HIMEM.SYS
DEVICE=EMM386.EXE
DOS=HIGH,UMB
```

Weet u zeker dat u de geheugenuitbreiding niet wilt gebruiken als EMS-geheugen (expanded memory), dan kunt u ook het geheugendeel nog gebruiken dat wordt gereserveerd voor de *page frame*:

```
DEVICE=HIMEM.SYS
DEVICE=EMM386.EXE NOEMS
DOS=HIGH,UMB
```

Behalve dat dit u upper memory oplevert, kunt u er het best gebruik van maken door device drivers op te slaan in dit gebied. Gebruik daarvoor de beide nieuwe opdrachten LOADHIGH en DEVICEHIGH. Overal waar u in uw CONFIG.SYS device drivers laadt met DEVICE, vervangt u die opdracht door DEVICEHIGH. In het bestand AUTOEXEC.BAT hebt u wellicht programma's als DOSKEY of KEYB resi-

dent geladen door eenvoudigweg de naam van het programma te typen. Plaats daar dan nu LOADHIGH of LH voor.

Hoe het geheugen wordt gebruikt, is goed te zien in het overzichtje dat de opdracht MEM/C op uw scherm plaatst. Het onderstaande voorbeeld geeft het gebruik van het werkgeheugen weer van een systeem met 4Mb geheugen, waarin naast de drie hierboven genoemde device drivers nog een paar drivers en TSR-programma's zijn geïnstalleerd.

```
Conventional Memory :

  Name                Size in Decimal        Size in Hex
-------------       ---------------------   -------------

  MSDOS               12864      ( 12.6K)       3240
  HIMEM                1184      (  1.2K)        4A0
  COMMAND              3008      (  2.9K)        BC0
  FREE               638128      (623.2K)      9BCB0

Total  FREE :        638128      (623.2K)

Total bytes available to programs :            638128    (623.2K)
Largest executable program size :              638128    (623.2K)
```

Toelichting

Voor het gebruik van upper memory bestaan geen standaardoplossingen. Veel hangt af van het gebruikte systeem, zowel voor wat betreft de hardware als de software. Zonder een goed begrip van zaken is een goed geheugenbeheer eigenlijk niet mogelijk.

Het upper memory maakt gebruik van het gebied tussen 640 Kb en 1024 Kb. Normaal gesproken is dat gereserveerd voor allerlei drivers, waar het besturingssysteem niet zonder kan. Maar op elk systeem worden gedeelten van het gebied niet gebruikt.

Alweer normaal gesproken, kan dit gebied echter niet als werkgeheugen worden geadresseerd. De driver EMM386.EXE spoort de vrije gebieden op en maakt ze toegankelijk. De opdracht DOS=UMB laat DOS dan weten dat van die gebieden gebruik mag worden gemaakt.

Wees echter voorzichtig met het upper memory: veel programma's kunnen er niet

mee overweg. Als ze geschreven zijn in de tijd dat MS-DOS 5.0 er nog niet was, is de kans groot, dat ze er zonder meer van uitgaan dat er niet meer dan 640 Kb werkgeheugen voorhanden is. Probeert u ze toch 'hoog' te laden, dan maken ze fouten of zorgen er zelfs voor dat het systeem blijft hangen. U dient dus even zelf uit te zoeken welke programma's het upper memory kunnen gebruiken.

Een bijzonderheid is nog dat expanded memory het beschikbare upper memory met 64 Kb vermindert. Expanded memory installeert namelijk een page frame van 64 Kb in het upper memory-gebied. Die page frame wordt gebruikt bij het *swappen* van programma's.

7.4 Geheugenvriendelijke systeembestanden

In de vorige paragraaf hebben we een aantal vrij algemene aanwijzingen gegeven voor een optimaal geheugenbeheer. Er blijven echter problemen te over. Vooral wanneer uw computer is aangesloten op een netwerk, verliest u nogal wat meer geheugen, want ook de netwerk-driver eist weer zijn deel daarvan op. In deze paragraaf zullen we voor een paar systemen de bestanden AUTOEXEC.BAT en CONFIG.SYS bespreken en zo inrichten dat ze zo veel mogelijk geheugen vrij houden.

Uitwerking

De bestanden AUTOEXEC.BAT en CONFIG.SYS zijn zo direct gekoppeld aan de PC zelf en de manier waarop u hem gebruikt, dat we hooguit algemene aanwijzingen kunnen geven. Terwijl de maatregelen uit de vorige paragraaf eigenlijk meer ten doel hadden geen geheugen ongebruikt te laten, gaat het nu om maatregelen die werkelijk geheugenruimte besparen.

Een XT zonder geheugenuitbreiding

De enige manier om op een XT met niet meer dan 640 Kb geheugen zo veel mogelijk geheugen beschikbaar te houden voor applicaties is het beperken van systeeminstellingen. De beide onderstaande bestanden zorgen samen voor een dergelijke minimale instelling.

CONFIG.SYS

```
REM CONFIG.SYS voor een XT met 640 Kb geheugen
SHELL=C:\DOS5\COMMAND.COM /P
COUNTRY=031,,C:\DOS5\COUNTRY.SYS
```

```
FILES=20
BUFFERS=20
BREAK=ON
STACKS=0,0
```

AUTOEXEC.BAT

```
REM AUTOEXEC.BAT voor een XT met 640 Kb geheugen
PATH C:\BAT;C:\DOS5
PROMPT $P$G
SET COMSPEC=C:\DOS5\COMMAND.COM
SET TEMP=C:\TEMP
SET TMP=C:\TMP
DATE
TIME
VER
```

Het kan zijn dat uw XT beschikt over een systeemklok die op een batterij loopt. In dat geval kunt u de opdrachten DATE en TIME achterwege laten.

Met deze configuratie beschikt u over ongeveer 571 Kb werkgeheugen onder MS-DOS 5.0. Daarmee kunnen vrijwel alle applicaties uit de voeten.

Een AT (80286) zonder geheugenuitbreiding

In principe zien de beide systeembestanden er voor een computer van het type AT met niet meer dan 640 Kb er precies zo uit als die voor een XT. Gebruik in ieder geval de programma's HIMEM.SYS en EMM386.EXE niet, omdat ze niet werken of zelfs geheugen vereisen in het werkgeheugen. Een AT is altijd uitgerust met een systeemklok, zodat de opdrachten DATE en TIME kunnen vervallen. Weliswaar levert dat geen geheugenwinst op, maar de startprocedure wordt wat rustiger en verloopt iets sneller.

Een AT (80286) met 1 Mb RAM-geheugen

Een typische AT, zoals hij het meest wordt verkocht, beschikt onder MS-DOS over 640 Kb rechtstreeks adresseerbaar geheugen en 384 Kb geheugenuitbreiding. Het is vooral belangrijk die 384 Kb zo effectief mogelijk te gebruiken.

Tip: De eerste AT's die op de markt kwamen, kunnen de geheugenuitbreiding alleen adresseren als het systeem het geheugen verdeelt in 512 Kb werk-

geheugen en 512 Kb extended memory. Is een dergelijke AT geconfigureerd met een werkgeheugen van 640 Kb, dan zijn de overige 384 Kb niet beschikbaar. Wendt u in dat geval tot uw dealer en vraag hem om oplossingen voor het probleem.

Beide onderstaande systeembestanden garanderen een optimaal gebruik van het totale geheugen.

CONFIG.SYS

```
REM CONFIG.SYS voor een AT (80286) met 1 Mb geheugen
DEVICE=C:\DOS5\HIMEM.SYS
SHELL=C:\DOS5\COMMAND.COM E:300 /P
COUNTRY=031,,C:\DOS5\COUNTRY.SYS
FILES=20
BUFFERS=20
BREAK=ON
STACKS=0,0
DOS=HIGH
```

AUTOEXEC.BAT

```
REM AUTOEXEC.BAT voor een AT (80286) met 1 Mb geheugen
PATH C:\BAT;C:\DOS5
PROMPT $P$G
SET COMSPEC=C:\DOS5\COMMAND.COM
SET TEMP=C:\TEMP
SET TMP=C:\TMP
VER
```

In deze configuratie levert het systeem u onder MS-DOS 5.0 ongeveer 618 Kb beschikbaar werkgeheugen. De opdracht MEM meldt dat DOS in de High Memory Area is geladen en dat er nog 327 680 bytes XMS-geheugen beschikbaar is. Dat geheugen kunt u gebruiken voor applicaties. Onder MS-DOS 5.0 kunt u er bijvoorbeeld een RAM-disk inrichten met de device driver RAMDRIVE.SYS of een cache-geheugen met de driver SMARTDRV.SYS. De beide drivers gebruiken dan wel weer een paar Kb van het werkgeheugen.

CONFIG.SYS met RAM-disk

```
REM CONFIG.SYS voor een AT (80286) met 1 Mb geheugen
REM XMS-geheugen -> RAM-disk
DEVICE=C:\DOS5\HIMEM.SYS
SHELL=C:\DOS5\COMMAND.COM E:300 /P
COUNTRY=031,,C:\DOS5\COUNTRY.SYS
```

```
FILES=20
BUFFERS=20
BREAK=ON
STACKS=0,0
DOS=HIGH
DEVICE=C:\DOS5\RAMDRIVE.SYS 320 /E
```

Tip: Vergeet u bij de opdracht om een RAM-disk te installeren de parameter /E, dan komt u van een koude kermis thuis. MS-DOS installeert dan namelijk de RAM-disk niet in het XMS-geheugen, maar in het conventionele werkgeheugen. U houdt dan nog maar een honderd Kb over en plotseling willen allerlei applicaties niet meer draaien.

Het systeem kent de nieuwe drive een eigen letter toe. In het gebruik onderscheidt deze drive zich maar op één manier van de gebruikelijke drives (A:, B:, C:, enzovoort): zet u de computer uit, dan zijn alle gegevens van de RAM-disk weg! Voor het overige kunt u er bestanden opslaan, kopiëren, wissen en alles wat u maar wilt. Het grote voordeel is de aanzienlijke tijdwinst bij lees- en schrijfhandelingen, terwijl de installatie u maar iets meer dan één Kb kost.

CONFIG.SYS met cache-geheugen

```
REM CONFIG.SYS voor een AT (80286) met 1 Mb geheugen
REM XMS-geheugen -> Cache
DEVICE=C:\DOS5\HIMEM.SYS
SHELL=C:\DOS5\COMMAND.COM E:300 /P
COUNTRY=031,,C:\DOS5\COUNTRY.SYS
FILES=20
BUFFERS=20
BREAK=ON
STACKS=0,0
DOS=HIGH
DEVICE=C:\DOS5\SMARTDRV.SYS 320
```

Gebruikt u bovenstaand bestand CONFIG.SYS, dan maakt de driver SMARTDRV.SYS van 320 Kb van het XMS-geheugen een cache-geheugen. Dat is een buffer die het besturingssysteem gebruikt bij lees- en schrijfhandelingen naar schijf. Geeft u de opdracht een bestand van schijf te lezen, dan kijkt het besturingssysteem eerst of de gegevens zich misschien in de buffer bevinden. Is dat zo, dan worden ze met elektronische in plaats van mechanische snelheid beschikbaar gemaakt. Geeft u de opdracht een bestand naar schijf te schrijven, dan gebeurt nagenoeg hetzelfde in omgekeerde richting. Installatie van een cache-geheugen met SMARTDRV.SYS kost u wel meer werkgeheugen dan een RAM-disk, namelijk 14 Kb.

Een 80386 met 1 Mb RAM-geheugen

Een standaard 80386SX-computer wordt uitgerust met 1 Mb RAM-geheugen. Bij het configureren is het belangrijk zoveel mogelijk systeemfuncties en device drivers in het extra geheugen te laden.

CONFIG.SYS

```
REM CONFIG.SYS voor een 80386 met 1 Mb geheugen
DEVICE=C:\DOS5\HIMEM.SYS
DEVICE=C:\DOS5\EMM386.EXE NOEMS
SHELL=C:\DOS5\COMMAND.COM E:300 /P
COUNTRY=031,,C:\DOS5\COUNTRY.SYS
FILES=20
BUFFERS=20
BREAK=ON
STACKS=0,0
DOS=HIGH,UMB
```

AUTOEXEC.BAT

```
REM AUTOEXEC.BAT voor een 80386 met 1 Mb geheugen
PATH C:\BAT;C:\DOS5
PROMPT $P$G
SET COMSPEC=C:\DOS5\COMMAND.COM
SET TEMP=C:\TEMP
SET TMP=C:\TMP
LOADHIGH DOSKEY
VER
```

Deze configuratie levert u onder MS-DOS 5.0 ongeveer 615 Kb (630320 bytes) werkgeheugen op. Dat is ongeveer even veel als bij een AT met 1 Mb geheugen. Het verschil zit echter in het feit dat nu al een aantal belangrijke drivers is geladen.

Dat komt omdat de 80386 in deze configuratie beschikt over upper memory, toegankelijk gemaakt door de driver EMM386.EXE. Roept u de driver EMM386.EXE op vanaf de systeemprompt, dan meldt het systeem:

```
MICROSOFT EXPANDED MEMORY MANAGER 386 Version 4.20.06X
(C) Copyright Microsoft Corporation 1986, 1991

Expanded Memory Services not available
```

```
Total upper memory available  ..             0K
Largest Upper Memory Block available ..      0K
Upper Memory Starting address ..          C800H

EMM386 active
```

Met de opdracht MEM/C krijgt u een overzicht van de verdeling van het geheugen:

```
Upper Memory :

   Name               Size in Decimal       Size in Hex
-------------     ---------------------   -------------
   SYSTEM           172064       (168.0K)     2A020
   KEYB               6208       (  6.1K)      1840
   DOSKEY             4144       (  4.0K)      1030
   FREE                128       (  0.1K)        80
   FREE              32560       ( 31.8K)      7F30
   FREE              46928       ( 45.8K)      B750

Total  FREE :        79616       ( 77.8K)

Total bytes available to programs (Conventional+Upper):      709936 (693.3K)
Largest executable program size :                            630128 (615.4K)
Largest available upper memory block:                         46928 ( 45.8K)

            1048576 bytes total contiguous extended memory
                  0 bytes available contiguous extended memory
             805888 bytes available XMS memory
                    MS-DOS resident in High Memory Area
```

Het grootste deel van het besturingssysteem bevindt zich in het high memory. Een klein gedeelte bevindt zich samen met HIMEM, EMM386 en een rest van COMMAND.COM in het werkgeheugen. In het upper memory vindt u behalve het grootste deel van het besturingssysteem, ook DOSKEY. In tegenstelling tot een AT pleegt deze configuratie geen aanslag op het conventionele werkgeheugen. In het upper memory is nog veel geheugen beschikbaar waarin u drivers of TSR-programma's kunt laden zonder dat dat ten koste gaat van het werkgeheugen. U gebruikt daarvoor de opdrachten DEVICEHIGH en LOADHIGH.

Een 80386 met 2 Mb RAM-geheugen of meer

In een 80386 met 1 Mb geheugen is na installatie van upper memory en high memory vrijwel geen extended memory meer beschikbaar als XMS-geheugen. Het gebruik daarvan is pas aan de orde, als de computer beschikt over meer geheugen.

In de configuratie die we hierboven bespraken, is wel XMS-geheugen beschikbaar. Dat komt omdat we daar het programma EMS386 hebben gestart met de optie NOEMS. EMS386.EXE richt in dat geval geen expanded memory in, zodat in het gebied tussen 640 Kb en 1 Mb geen ruimte hoeft te worden gereserveerd voor de page frame. Werkt u met meer geheugen en met programma's die expanded memory gebruiken, dan laat u ide de optie NOEMS weg.

Device drivers en programma's in de juiste volgorde plaatsen

U kunt nogal wat geheugenruimte sparen door device drivers en programma's die in het upper memory kunnen worden geladen, in een uitgekiende volgorde op te nemen in de bestanden CONFIG.SYS en AUTOEXEC.BAT. Roep eerst met de opdracht MEM/C een overzicht van de geheugenindeling op uw scherm.

Als eerste wordt bijvoorbeeld een relatief klein programma, KEYB, in het upper memory geladen, dat een deel van een vrij groot UMB-block beslaat. Een beduidend groter programma, bijvoorbeeld een netwerk-driver IPX, kan dan niet meer in het upper memory worden geladen. Het programma KEYB is maar zo'n 6 Kb groot; IPX zo'n 17 Kb.

Door de volgorde waarin beide programma's worden geladen om te draaien, bestaat de kans dat er wel een UMB-block is waar IPX in past. Ook al heeft dat tot gevolg dat KEYB in het gewone werkgeheugen moet worden geladen, dan nog blijft daar 11 Kb geheugen meer beschikbaar.

7.5 MS-DOS 5.0 en Windows 3.0 combineren

De kans dat u op een 80386 problemen krijgt met Windows nadat u het systeem hebt geconfigureerd zoals we hebben besproken in de vorige paragraaf, is niet gering. Windows draait niet meer of schakelt over naar *real mode*. Voordat we die problemen kunnen oplossen, moeten we wat meer vertellen over Windows.

Uitwerking

Windows 3.0 is al eerder dan MS-DOS 5.0 uitgerust met functies om het geheugen optimaal te gebruiken. Ook Windows gebruikt bijvoorbeeld high memory en upper memory.

Daarnaast kan Windows in drie verschillende modi draaien, die alle drie op een andere manier gebruik maken van het geheugen.

Windows in de real mode

In de real mode is Windows 3.0 compatibel met oudere versies, die gedeeltelijk alleen in real mode werken.

In de real mode kan Windows alleen uit de voeten met expanded memory. U dient dan voor een optimale samenwerking tussen MS-DOS en Windows alle geheugenruimte van de geheugenuitbreiding met behulp van EMM386.EXE tot expanded memory te maken. Gebruik daarvoor de optie RAM of laat de optie NOEMS achterwege.

Bovendien kunt u, als u beschikt over voldoende geheugen, een cache inrichten voor de harde schijf met SMARTDRV. Het bestand CONFIG.SYS voor een computer met 4 Mb geheugen en Windows in de real mode zou er als volgt uit kunnen zien:

CONFIG.SYS

```
REM CONFIG.SYS voor een 80386 met 4 Mb geheugen
REM en Windows in real mode
DEVICE=C:\DOS5\HIMEM.SYS
DEVICE=C:\DOS5\EMM386.EXE
SHELL=C:\DOS5\COMMAND.COM E:300 /P
COUNTRY=031,,C:\DOS5\COUNTRY.SYS
FILES=20
BUFFERS=3
DEVICE=C:\DOS5\SMARTDRV.SYS 512 /A
BREAK=ON
STACKS=0,0
DOS=HIGH,UMB
```

Voor de cache is 512 Kb van het expanded memory gereserveerd. Bij het gebruik van een cache zijn nog maar weinig buffers nodig. Drie is genoeg; meer buffers staat zelfs goed functioneren van de cache in de weg.

Windows in de standard mode

Als Windows in de standard mode draait, kunt u geen expanded memory gebruiken. U dient dan dus achter de oproep van EMM386.EXE de optie NOEMS toe te voegen. Als u beschikt over genoeg geheugen, kunt u de cache met SMARTDRV installeren zonder de optie /A. Dat heeft tot gevolg dat XMS-geheugen wordt gebruikt voor de cache.

Windows in de protected mode

Voordat Windows in de protected of enhanced mode kan draaien, moet aan een paar voorwaarden zijn voldaan. In de eerste plaats moet de speciale driver WINA20.386 beschikbaar zijn in de hoofddirectory van de harde schijf van waar het systeem wordt gestart.

In principe wordt het systeem vervolgens geconfigureerd zoals in de standard mode, dus zonder expanded memory. Bij de installatie van SMARTDRV moet u een minimale grootte aangeven, omdat Windows anders al gauw de hele cache opeist voor eigen gebruik, met alle consequenties vandien: u vraagt zich af of uw harde schijf niet meer deugt, zo traag gaat alles.

CONFIG.SYS

```
REM CONFIG.SYS voor een 80386 met 4 Mb geheugen
REM en Windows in de protected mode
DEVICE=C:\DOS5\HIMEM.SYS
DEVICE=C:\DOS5\EMM386.EXE NOEMS
SHELL=C:\DOS5\COMMAND.COM E:300 /P
COUNTRY=031,,C:\DOS5\COUNTRY.SYS
FILES=20
BUFFERS=3
DEVICE=C:\DOS5\SMARTDRV.SYS 512 256
BREAK=ON
STACKS=0,0
DOS=HIGH,UMB
```

7.6 Storingen verhelpen bij geheugenbeheer

Als u uitgebreid aan het werk gaat om uw geheugen zo optimaal mogelijk in te richten, zal er vast nog wel wat fout gaan onderweg. Zo lang de problemen beperkt blijven tot foutmeldingen, is er nog niet zo veel aan de hand, maar wanneer een fout in

het bestand CONFIG.SYS een crash van het systeem veroorzaakt, kan het best zijn dat u het systeem niet meer vanaf de harde schijf kunt starten. We zullen in deze paragraaf een aantal mogelijke voorzorgsmaatregelen en oplossingen voor deze problemen bespreken.

Uitwerking

In de allereerste plaats dient u een systeemdiskette aan te maken. Plaats een nieuwe diskette in drive A: en geef de opdracht:

```
FORMAT A:/S
```

Hebt u een al geformatteerde diskette met genoeg ruimte bij de hand, dan kunt u daarop de systeembestanden kopiëren met

```
SYS A:
```

Loopt nu uw systeem vast bij de uitvoering van het bestand CONFIG.SYS, dan kunt u altijd weer starten met de systeemdiskette in A:.

Als een crash wordt veroorzaakt door een fout in het bestand AUTOEXEC.BAT, kunt u opnieuw starten vanaf de harde schijf. Druk dan direct op CTRL-C, waardoor het bestand AUTOEXEC.BAT wordt onderbroken voor uitvoering van de eerste opdracht. Vanzelfsprekend beschikt u dan nog niet over een zoekpad en soortgelijke functies.

De systeemdiskette werkt het best als u er behalve de systeembestanden van DOS ook de andere configuratie-bestanden naar toe kopieert. In dat geval kunt u ze namelijk altijd terugkopiëren naar uw harde schijf om de foutieve versies die de crash hebben veroorzaakt, te vervangen. Sla de bestanden CONFIG.SYS en AUTOEXEC.BAT van de harde schijf niet op de systeemdiskette op onder hun eigen naam, omdat MS-DOS ze dan bij het starten zal gebruiken. Als het in dat geval een experimentele versie betreft die een systeemfout heeft veroorzaakt, zal ook de systeemdiskette uw computer niet meer aan de praat krijgen.

Sla ze dus op onder een naam met een afwijkende extensie, bijvoorbeeld CONFIG.LWC en AUTOEXEC.LWC. Gebruik niet de extensies BAK en SIK, want die worden door tekstverwerkingsprogramma's gebruikt.

U doet er verstandig aan naast die beide bestanden ook de bestanden COUNTRY.SYS, DOSKEY.COM, EDIT.COM en QBASIC.EXE naar de systeemdiskette

te kopiëren. Ze worden gebruikt in de standaardbestanden CONFIG.SYS en AUTOEXEC.BAT die we hieronder bespreken. In plaats van EDIT.COM en QBASIC.EXE kunt u natuurlijk ook de minder krachtige, maar ook minder omvangrijke regel-editor EDLIN naar de systeemdiskette kopiëren.

Voor het starten van het systeem vanaf de systeemdiskette kunt u het best de volgende bestanden CONFIG.SYS en AUTOEXEC.BAT op de diskette plaatsen:

CONFIG.SYS

```
REM CONFIG.SYS voor een systeemdiskette
COUNTRY=031,,C:\DOS5\COUNTRY.SYS
FILES=20
BUFFERS=20
BREAK=ON
STACKS=0,0
```

AUTOEXEC.BAT

```
REM AUTOEXEC.BAT voor een systeemdiskette
PROMPT $P$G
DOSKEY
VER
```

Andere belangrijke bestanden

Afhankelijk van de configuratie van het systeem heeft het zin nog meer drivers en programma's naar uw systeemdiskette te kopiëren. Loop even de bestanden CONFIG.SYS en AUTOEXEC.BAT langs om na te gaan welke bestanden dat zijn.

Daarnaast is het nog belangrijk de hoofddirectory van de harde schijf onder de loep te nemen. Windows 3.0 verwacht bijvoorbeeld het bestand WINA20.386 in de hoofddirectory van de schijf waarmee het systeem wordt gestart. Dat programma dient dus op de systeemdiskette te staan als u het normaal gesproken gebruikt.

Problemen met het versienummer

Als u de hierboven beschreven systeembestanden gebruikt, kan het gebeuren dat applicaties die een bepaald versienummer van DOS vereisen, niet meer willen draaien. U dient in dat geval het bestand CONFIG.SYS aan te passen.

Uitwerking

Onder MS-DOS 5.0 kunt u applicaties voor de gek houden voor wat betreft het versienummer. Dat doet u met de opdracht SETVER. De aanpassing bestaat uit twee delen: in de eerste plaats moet de driver steeds in het bestand CONFIG.SYS worden geactiveerd en in de tweede plaats moet de versietabel in de verborgen systeembestanden worden voorzien van het nieuwe versienummer.

Voor het maken van een systeemdiskette heeft dat de volgende consequenties:

1 De DOS-opdracht SETVER moet op de systeemdiskette staan.

2 In het bestand CONFIG.SYS moet de volgende opdracht staan:

```
DEVICE=SETVER.EXE
```

3 U moet de versietabel van de diskette actualiseren. Dat doet u door de opdracht SETVER op te roepen of vanaf de harde schijf de opdracht SYS A: te geven. In het laatste geval gaan we ervan uit dat de versietabel van de harde schijf de juiste gegevens bevat.

De foutmelding Packed File Corrupt

U hebt MS-DOS netjes weggestopt en zodoende meer dan 620 Kb werkgeheugen gecreëerd, maar sommige programma's geven de foutmelding *Packed File Corrupt.* Dat programma kan dan niet uit de voeten met het geheugenbeheer van MS-DOS 5.0. Met behulp van programma LOADFIX kunt u dit probleem oplossen.

Uitwerking

Als bijvoorbeeld de editor EDIT niet goed functioneert en de hierboven beschreven foutmelding geeft, kunt u met de volgende opdracht het programma toch starten:

```
LOADFIX EDIT
```

Het effect is dat de eerste 64 Kb van het geheugen niet voor EDIT wordt gebruikt, waardoor het probleem zich niet langer voordoet.

8 Alle bestanden binnen bereik

Normaal gesproken voert DOS alleen opdrachten uit in de actuele directory of in een opgegeven directory. Dat maakt het soms nogal lastig om een opdracht uit te voeren voor een aantal bestanden in meer dan één directory. In dit hoofdstuk zullen we u een aantal tips aan de hand doen waarmee u dit probleem te lijf kunt.

8.1 Opdrachten uitvoeren voor alle bestanden van een schijf

MS-DOS is niet in staat om een opdracht uit te voeren voor de bestanden van een directory én de bestanden in de subdirectories. Het zou bijvoorbeeld heel handig zijn als u alle bestanden op een schijf met de extensie .TMP (tijdelijke bestanden) in één keer kon wissen. Of als u in één keer alle bestanden zou kunnen doorzoeken naar een bepaald begrip. De programma's die we hieronder bespreken hebben tot doel dat alles toch mogelijk te maken.

Uitwerking

Met de beide programma's die volgen kunt u nagenoeg alle DOS-opdrachten die daarvoor in aanmerking komen, in één keer uitvoeren voor alle bestanden. Het gaat om een batchbestand en een klein programma in QBASIC. Het batchbestand zoekt alle passende bestandsnamen en schrijft die naar een tijdelijk bestand. Het BASIC-programma genereert uit die namen de juiste DOS-opdrachten en plaatst die in een nieuw batchbestand, waarmee aan het doel is voldaan.

Het batchbestand FINDALL.BAT

U dient het batchbestand FINDALL.BAT op te slaan in dezelfde directory als het BASIC-programma of in een directory die deel uitmaakt van een zoekpad. Hebt u op uw harde schijf een directory C:\BAT ingericht, dan hoort ook dit batchbestand daar thuis. Het ziet er als volgt uit:

```
@REM *****************************************************
@REM * Dit batchbestand maakt een lijst van een selectie *
@REM * van bestanden uit alle directories van een schijf *
```

```
@REM ****************************************************
@ECHO OFF
IF "%1" == "" GOTO Syntaxis
CHKDSK /v | FIND "\" | FIND /v "Directory" | FIND /I "%1"
                                                              >ALLBST.DAT
GOTO Einde
:Syntaxis
ECHO Dit batchbestand maakt een lijst van bestanden
ECHO Syntaxis: FINDALL BESTAND.EXT
ECHO (delen van namen wel, wildcards niet toegestaan)
:Einde
@ECHO ON
```

Bij het oproepen van het batchbestand voert u een bestandsnaam of een deel daarvan in. Alle bestandsnamen die aan het criterium voldoen, worden dan naar het bestand ALLBST.DAT geschreven. Dat bestand dient als invoer voor het BASIC-programma. Wildcards zijn niet toegestaan, maar ook zonder wildcards, namelijk door een deel van de bestandsnaam als parameter mee te geven aan de opdracht, kunt u bestanden selecteren.

Een paar voorbeelden:

FINDALL .TMP — Er wordt een lijst gemaakt met alle tijdelijke bestanden.

FINDALL .BAS — Alle programma's met de extensie .BAS, standaard-extensie voor programma's gemaakt in BASIC, worden opgeslagen in ALLBST.DAT. Zou u in dit geval de punt weglaten, dan nam het batchbestand ook bestandsnamen als QBASIC.EXE op in de lijst. Het is dus oppassen geblazen.

FINDALL COMMAND.COM — Alle geïnstalleerde versies van het besturingssysteem worden verzameld.

Tip: FINDALL zoekt en vindt ook probleemloos verborgen systeembestanden. Ook attributen als Read only zijn geen belemmering voor FINDALL. Dit kan in het verdere verloop voor problemen zorgen als de DOS-opdracht die uiteindelijk moet worden uitgevoerd niet met systeembestanden of een dergelijk attribuut overweg kan.

Het BASIC-programma DOALL.BAS

We hebben het onderstaande programma zo kort mogelijk gehouden om u een omvangrijke typeklus te besparen.

```
REM *************************************************************
REM *                        DOALL.BAS                          *
REM * voert een DOS-opdracht uit voor een selectie van bestanden *
REM *                      van een schijf                       *
REM *************************************************************
REM
REM Invoer opvragen
CLS : PRINT : PRINT
PRINT "Typ een deel van de bestandsnaam in HOOFDLETTERS"
PRINT "Geef bij extensies ook de punt op, bijvoorbeeld: .TMP"
INPUT "Om welke bestanden gaat het"; BST$
PRINT "Welke DOS-opdracht moet worden uitgevoerd?"
LINE INPUT OPD$
PRINT "Geef hieronder parameters of doel aan:"
LINE INPUT PARA$
INPUT "Wilt u de functie ECHO inschakelen (J/N)"; ECHO$
REM
REM batchbestand FINDALL.BAT oproepen om de bestanden te zoeken
OPDRACHT$ = "FINDALL.BAT " + BST$
PRINT OPDRACHT$
SHELL OPDRACHT$
REM
REM Eerste bestand lezen en opdracht construeren
AANTAL = 1
OPEN "ALLBST.DAT" FOR INPUT AS #1
  OPEN "DOALL!!!.BAT" FOR OUTPUT AS #2
    CLS
    PRINT "Het batchbestand DOALL!!!.BAT
    PRINT "bevat de volgende opdrachten:"
    IF ECHO$ ="n" OR ECHO$ = "N" THEN PRINT #2, "@ECHO OFF"
    PRINT "@ECHO OFF"
    WHILE NOT EOF (1)
      INPUT #1, INVOER$
      UITVOER$ = OPD$ + " " + INVOER$ + " " + PARA$
      PRINT UITVOER$
      PRINT #2, UITVOER$
```

```
        AANTAL = AANTAL + 1
        IF AANTAL >= 20 THEN
          AANTAL = 1
          PRINT "Druk op een toets..."
          WHILE INKEY$ = "" : WEND
        END IF
      WEND
      IF ECHO$ ="n" OR ECHO$ = "N" THEN PRINT #2, "@ECHO ON"
      PRINT "@ECHO ON"
    CLOSE #2
  CLOSE #1
  PRINT "Het batchbestand DOALL!!!.BAT voert de opdracht uit"
  PRINT "voor alle bestanden van deze lijst. Controleer de lijst"
  PRINT "voordat u het batchbestand activeert!"
  PRINT "Druk op een toets om het programma te beëindigen."
  WHILE INKEY$ = "" : WEND
  CLS
  END
```

Tip: Vergeet niet dat u bij de opgave van de bestandsnamen geen wildcards mag gebruiken. Voer in plaats van *.TXT gewoon .TXT in. De laatste opdracht van het programma:

```
END
```

mag u ook vervangen door:

```
SYSTEM
```

BASIC wordt dan automatisch verlaten; u keert terug naar het besturingssysteem en de DOS-prompt verschijnt weer op uw scherm.

Om de werking van het programma te illustreren gebruiken we de opdracht DIR. Die is volstrekt ongevaarlijk, maar laat wel zien wat het programma kan. We willen een overzicht van alle bestanden COMMAND.COM van de harde schijf (C:) inclusief omvang, datum en tijd.

Om te beginnen laadt u het programma QBASIC en aansluitend het programma DOALL.BAS. Start het programma met Shift-F5.

1 Het programma vraagt u op welke bestanden de opdracht moet worden uitgevoerd. U typt COMMAND.COM.
2 Het programma vraagt u welke DOS-opdracht moet worden uitgevoerd. U typt DIR.

3 Het programma vraagt vervolgens of u nog parameters wilt opgeven om de opdracht onder bepaalde opties uit te voeren. In ons geval is dat niet nodig. U hoeft slechts op Enter te drukken. Zou u bestanden willen kopiëren, dan kon u hier het doelstation opgeven.

4 Het programma wil ten slotte weten of de functie ECHO moet worden ingeschakeld. Typ J, zodat u het verdere verloop op het scherm kunt volgen.

5 Het BASIC-programma roept nu het eerder aangemaakte batchbestand op samen met de door u ingevoerde bestandsnaam. De opdracht daartoe wordt op het scherm weergegeven. In ons geval luidt die:

```
FINDALL COMMAND.COM
```

6 Als het batchbestand alle in aanmerking komende bestanden gevonden heeft, maakt het BASIC-programma voor elk daarvan de DOS-opdracht aan en schrijft die naar een nieuw batchbestand DOALL!!!.BAT. De ietwat ongebruikelijke naam moet verhinderen dat u het batchbestand per ongeluk activeert.

7 Alle aangemaakte opdrachten worden per pagina op uw scherm getoond, zodat u kunt controleren of u geen fouten hebt gemaakt.

8 Tot slot herinnert het BASIC-programma u aan de naam van het nieuw gemaakte batchbestand en waarschuwt u nog een keer. Hoewel ons batchbestand rond de opdracht DIR ongevaarlijk is, zullen we het nog een keer bekijken voor alle zekerheid.

8 Beëindig QBASIC en laad het bestand DOALL!!!.BAT in een tekstverwerker. De veiligste methode om het bestand te bekijken is met de opdracht:

```
TYPE DOALL!!!.BAT | MORE
```

Het resultaat kan er als volgt uitzien:

```
DIR C:\COMMAND.COM
DIR C:\DOS5\COMMAND.COM
DIR C:\BEVEIL\COMMAND.COM
```

10 Activeer nu het batchbestand door de naam achter de systeemprompt te typen. Een lijstje met alle versies van COMMAND.COM verschijnt op uw scherm.

Toelichting

FINDALL maakt gebruik van de DOS-opdracht CHKDSK met de optie /V. Die zorgt voor uitvoer van alle bestanden van een schijf met zoekpad. De beide eerste opdrachten FIND filteren alle overbodige gegevens zoals drive, capaciteit en opgave van directories uit de lijst. De laatste opdracht FIND zoekt de bestanden uit die voldoen aan het opgegeven criterium. Het resultaat van die zoekactie wordt dan omgeleid naar ALLBST.DAT.

Het BASIC-programma DOALL.BAS vraagt eerst alle noodzakelijke parameters op. De DOS-opdracht (OPD$) en de eventuele parameters bij die opdracht (PARA$) worden ingevoerd met de BASIC-opdracht LINE INPUT, zodat u ook gebruik kunt maken van aanhalingstekens. De opdracht FIND doet dat bijvoorbeeld.

Daarna wordt het batchbestand FINDALL.BAT in werking gesteld om alle bestanden die aan de specificaties voldoen, te zoeken. Het resultaat van het batchbestand staat in het bestand ALLBST.DAT. Het bestand DOALL!!!.BAT wordt geopend om er de nieuwe opdrachten naar toe te schrijven. Als de functie ECHO uitgeschakeld is, wordt als eerste opdracht @ECHO OFF opgenomen in het batchbestand. De apestaart zorgt ervoor dat die opdracht zelf niet op het scherm verschijnt. Daarna worden stuk voor de stuk de bestandsnamen uit het ALLBST.DAT gelezen, gecombineerd met de DOS-opdracht en naar DOALL!!!.BAT geschreven. Ter controle wordt de nieuwe opdracht ook weergegeven op het scherm. Het programma houdt een regelteller AANTAL bij, zodat u bij een vol scherm de gelegenheid krijgt op een toets te drukken voordat het programma verder gaat.

Als het batchbestand is geopend met een @ECHO OFF, zal het worden gesloten met de opdracht @ECHO ON. Het BASIC-programma sluit af met een waarschuwing nog eens goed na te denken voordat u het batchbestand activeert. Sla de waarschuwing niet in de wind, een foutje kan catastrofale gevolgen hebben!

8.2 Selecteren op grond van meer criteria

U kunt met FINDALL.BAT al heel wat doen om bestanden te selecteren. Toch zou het mooi zijn als u niet alleen een selectie zou kunnen maken op grond van de extensie '.TXT', maar ook op grond van de beginletter van de bestandsnaam, bijvoorbeeld 'B'. We zullen daarvoor een tweede batchbestand gebruiken.

Uitwerking

Het probleem laat zich betrekkelijk eenvoudig oplossen. In eerste instantie maken we gewoon het batchbestand DOALL!!!.BAT op grond van de extensie .TXT. Om nu de bestanden die beginnen met de letter B eruit te filteren, gebruiken we het volgende batchbestand FINDMORE.BAT.

```
@REM **********************************
@REM * Dit batchbestand selecteert op *
@REM * grond van secundaire criteria  *
@REM **********************************
@ECHO OFF
IF "%1" == "" GOTO Syntaxis
IF EXIST ALLBST.OLD DEL ALLBST.OLD
RENAME ALLBST.DAT ALLBST.OLD
FIND <ALLBST.OLD >ALLBST.DAT /I "%1"
GOTO Einde
:Syntaxis
ECHO Dit batchbestand maakt een tweede selectie
ECHO Syntaxis: FINDALL BESTAND.EXT
ECHO (delen van namen wel, wildcards niet toegestaan)
:Einde
@ECHO ON
```

Sla het batchbestand op onder de naam FINDMORE.BAT in de directory waar u de beide andere bestanden ook hebt opgeslagen.

Om het programma te kunnen activeren moeten we het BASIC-programma aanpassen. Tussen de bestaande regels 'SHELL OPDRACHT$' en de erop volgende 'REM' dient u de onderstaande regels toe te voegen:

```
DO
  PRINT "Geef secundaire selectiecriteria op"
  LINE INPUT "of druk op Enter"; SELECT$
  IF SELECT$ <> "" THEN
    OPDRACHT$ = "FINDMORE.BAT " + SELECT$
    SHELL OPDRACHT$
  END IF
LOOP WHILE SELECT$ <> ""
```

We zullen nu het aangepaste programma testen. Start DOALL.BAS en voer de volgende gegevens in:

```
Bestandsnaam:                : \B
Opdracht                     : DIR
parameter                    :
ECHO                         : N
secundair selectiecriterium: .TXT
secundair selectiecriterium:
```

Het resultaat is een kleinere lijst met bestanden dan in de vorige paragraaf, omdat nu alleen bestanden met de beginletter B en de extensie .TXT worden geselecteerd.

Toelichting

Het nieuwe batchbestand FINDMORE.BAT onderzoekt met de opdracht IF EXIST of er een bestand ALLBST.OLD bestaat. Iedere keer dat het batchbestand wordt geactiveerd, blijft er namelijk een dergelijk bestand achter. Als het aanwezig is, wordt het gewist. Daarna wordt het bestand ALLBST.DAT, met alle bestandsnamen die in eerste instantie zijn geselecteerd, hernoemd tot ALLBST.OLD. De opdracht FIND selecteert uit die bestanden vervolgens alleen die bestanden die voldoen aan het tweede criterium.

Het BASIC-programma vraagt in de toegevoegde regels of er een tweede selectiecriterium moet worden toegepast. Voert u niets in, maar drukt u op Enter, dan gaat het programma verder zoals tot nu toe. Geeft u wel een criterium op, dan selecteert het programma daarop. De vraag naar een volgend criterium wordt net zo lang herhaald tot u geen nieuw criterium opgeeft.

De backslash voor de hoofdletter B zorgt ervoor dat alleen bestanden worden geselecteerd die beginnen met die letter. Zou u de backslash weglaten, dan zou het programma naast bestanden als ‘BEATLES.TXT’ ook ‘THE_BAND.TXT’ opnemen.

8.3 Zoeken naar een string

In paragraaf 6.8 hebben we een batchbestand beschreven met behulp waarvan u een string kunt zoeken in bestanden. De tekortkoming van dat batchbestand is dat alleen de bestanden van de opgegeven directory worden doorzocht. Het is echter mogelijk met behulp van de programma's uit de vorige paragraaf een string te zoeken in alle bestanden van de schijf.

Uitwerking

We zoeken in alle bestanden van de harde schijf naar de string 'WORD5'. U start het programma DOALL.BAS en voert de volgende gegevens in:

```
Bestandsnaam:               : .TXT
Opdracht                    : FIND "WORD5"
parameter                   :
ECHO                        : N
secundair selectiecriterium:
```

Het batchbestand dat wordt aangemaakt, bevat alle bestanden met de gezochte string. Start u het batchbestand DOALL!!!.BAT, dan worden ze getoond, steeds met de regel waarin de gezochte string voorkomt.

Toelichting

Wellicht vraagt u zich af waarom we bij de vraag naar de opdracht niet alleen FIND hebben getypt, maar ook een parameter, namelijk de zoekstring. Dat komt omdat de opdracht FIND eerst de zoekstring verwacht en dan pas de bestandsnaam. We hebben het programma bewust zo gemaakt dat in dit soort situaties is voorzien.

8.4 Alle bestanden van de harde schijf testen op integriteit

Soms lukt het plotseling niet meer een bestand te lezen. Betreft het een bestand op een diskette, dan kunt u zich daar vrij eenvoudig tegen beschermen door een reservekopie van de diskette te maken. Als DOS een bestand kan kopiëren, kan het dat ook vrijwel altijd lezen. Gaat het om een bestand op een harde schijf, dan is dat niet zo makkelijk. Een backup kost namelijk nogal wat tijd. Er is een snellere methode die gebruik maakt van het randapparaat NUL.

Uitwerking

Het niet-bestaande randapparaat NUL is een mogelijkheid om uitvoer om te leiden in het niets, in plaats van naar het scherm of naar een printer. Kopieert u bestanden naar NUL, dan bespaart u zich alle tijd die u anders kwijt bent aan schrijfhandelingen.

Gebruik het BASIC-programma DOALL.BAS en selecteer bestanden. Voer als opdracht COPY in en als parameter >NUL. Het batchbestand dat wordt aangemaakt, probeert alle geselecteerde bestanden te kopiëren naar NUL. Als dat geen foutmeldingen oplevert, weet u dat DOS de bestanden kan lezen.

8.5 Een selectie van bestanden wissen op de harde schijf

Sommige applicaties leggen bestanden aan, maar wissen die niet meer na gebruik. Vaak zijn het tijdelijke bestanden met extensies als .TMP, maar ook veel backupbestanden, met extensies als .BAK en .SIK, zijn na verloop van tijd overbodig. Wilt u die wissen, dan zult u normaal gesproken alle directories langs moeten gaan en steeds de opdracht DEL *.BAK of iets dergelijks moeten geven. Met de programma's van dit hoofdstuk kunt u zich die moeite besparen.

Uitwerking

Het kost met DOALL.BAS en FINDALL.BAT zo weinig moeite, dat u er wel even bij moet blijven! Voordat u het weet, hebt u zich veel extra werk op de hals gehaald, omdat u een verkeerde selectie maakt. Start het programma DOALL.BAS en geef als opdracht DEL. Bij de vraag naar de bestandsnaam geeft u bijvoorbeeld .TMP op. Het batchbestand dat wordt aangemaakt, is dan in staat alle tijdelijke bestanden op de harde schijf te wissen, ongeacht in welke directory ze staan.

Let op! Het programma wist echt alle bestanden die u hebt geselecteerd. Controleer het overzicht dat op uw scherm verschijnt dus zorgvuldig. Bent u niet zeker van uw zaak, wis dan het batchbestand DOALL!!!.BAT en wis overbodige bestanden stuk voor stuk handmatig.

8.6 Opdrachten toepassen op een groep directories

Soms is het een beetje te veel van het goede om een DOS-opdracht uit te voeren met betrekking tot de totale harde schijf. U hebt uw harde schijf ongetwijfeld voorzien van een keurige directory-structuur en zult dus vaak genoeg een opdracht willen uitvoeren in een bepaalde directory en de directories die daar direct aan ondergeschikt zijn. De oplossing heeft te maken met het zoekpad dat FINDALL gebruikt.

Uitwerking

Als FINDALL u vraagt de bestandsnaam in te voeren waarop moet worden geselecteerd, kunt u ook met behulp van backslashes directories opgeven. Hebt u bijvoorbeeld een directory C:\TEKSTEN met een aantal subdirectories, dan is het zoekpad van alle bestanden in die directories om te beginnen C:\TEKSTEN. Bij de vraag naar de bestandsnaam voert u dus in:

```
\TEKSTEN\
```

Wilt u in die directories bepaalde bestanden selecteren, dan zult u de mogelijkheid moeten gebruiken secundaire criteria te kiezen.

8.7 Bestanden alfabetisch sorteren met opgave van zoekpad

Op een harde schijf van 40 Mb die intensief wordt gebruikt, staan al gauw meer dan 1000 bestanden. Dat gaat ten koste van het overzicht. U kunt met de opdracht CHKDSK /V natuurlijk een overzicht van alle directories met alle bestanden op uw scherm opvragen, maar hieronder bespreken we een programma dat alle bestanden alfabetisch gesorteerd weergeeft inclusief het zoekpad. Dat maakt het u bijvoorbeeld mogelijk snel een antwoord te krijgen op de vraag of bepaalde bestanden meerdere keren op de harde schijf voorkomen.

Uitwerking

Maak het volgende programma SORTBST.BAS:

```
REM ***************************************************************
REM *                      SORTBST.BAS                            *
REM * sorteert de bestanden van een schijf met opgave van zoekpad *
REM *                                                             *
REM ***************************************************************
REM
REM gegevens opvragen
CLS : PRINT : PRINT
INPUT "Wilt u de uitvoer naar het scherm sturen of naar een bestand
                                                        (S/B)"; UIT$
IF UIT$ = "B" OR UIT$ = "b" THEN UIT$ = " | MORE"
```

```
IF UIT$ = "D" OR UIT$ = "d" THEN UIT$ = " >SORTBST.DAT"
SHELL "FINDALL.BAT " + "\"
OPEN "ALLBST.DAT" FOR INPUT AS #1
  OPEN "ALLBST.SRT" FOR OUTPUT AS #2
    WHILE NOT EOF (1)
      PRINT ".";
      INPUT #1, INVOER$
      L = LEN(INVOER$0
      FOR i = L TO 1 STEP -1
        IF MID$(INVOER$,i,1) = "\"
        THEN BSTNAAM$ = MID$(INVOER$,i+1,L-i) : i = 1
      NEXT i
      WHILE LEN(BSTNAAM$) > 15
        BSTNAAM$ = BSTNAAM$ + " "
      WEND
      PRINT #2, BSTNAAM$ + INVOER$
    WEND
  CLOSE #2
CLOSE #1
OPDRACHT$ = "SORT < ALLBST.SRT " + UIT$
SHELL OPDRACHT$
IF UIT$ = ">SORTBST.DAT"
THEN PRINT "Gesorteerde bestandenlijst: SORTBST.DAT"
END
```

Tip: De opdracht SORT van MS-DOS kan bij harde schijven met een grote capaciteit in problemen komen als een buffer nodig is van meer dan 64 Kb. Gebruik in dat geval de hieronder beschreven oplossing die gebruik maakt van de DOS Shell.

Het grote voordeel van het BASIC-programma is dat u de uitvoer kunt omleiden naar een bestand. Dat bestand kunt u dan in een tekstverwerker verder bewerken of gebruiken als invoer voor een ander programma.

De DOS Shell biedt de mogelijkheid uit de lijst een keuze van bestanden te maken, die u dan verder kunt bewerken. Daartoe start u de DOS Shell en kiest All files in het menu View. U kunt dan bestanden markeren en selecteren.

8.8 De bestanden van het zoekpad op een rijtje

Soms zijn er van die situaties waarin er niets fout gaat en er toch niet gebeurt wat u wilt. Uiteindelijk blijkt dan bijvoorbeeld dat van twee bijna identieke gegevensbestanden met dezelfde naam die allebei via het zoekpad gevonden kunnen worden, de verkeerde is gelezen. Het programma dat we in deze paragraaf bespreken, biedt u een overzicht van alle bestanden in het zoekpad. Het geeft u bovendien het overzicht dat u nodig hebt om het zoekpad te kunnen opschonen, wat nogal wat tijdwinst kan opleveren.

Uitwerking

Het batchbestand SHOWPATH.BAT geeft alle bestanden van het zoekpad alfabetisch gesorteerd weer. Daarnaast berekent het bestand hoeveel bestanden via het zoekpad kunnen worden gevonden. Voorwaar geen geringe prestatie voor een batchbestand. We raden u aan de device driver ANSI.SYS via het bestand CONFIG.SYS te installeren, voordat u het batchbestand activeert. Zonder die driver werkt het programmaatje weliswaar ook, maar de uitvoer naar het scherm ziet er dan minder fraai uit.

```
@ECHO OFF
REM SHOWPATH.BAT door Manfred Tornsdorf
SET OLDPROMPT=%PROMPT%
@IF EXIST SHOWPATH.TMP DEL SHOWPATH.TMP
FOR %%a IN (%PATH%) DO DIR %%a >>SHOWPATH.TMP
FIND "." SHOWPATH.TMP | FIND /V "." "---------" |SORT | MORE
FIND /C "." SHOWPATH.TMP
@ECHO ON
@PROMPT $e[3A

@ECHO Gevonden bestanden
@SET PROMPT=%OLDPROMPT%
@SET OLDPROMPT=
```

Het batchbestand zit vol met trucs die u ook in uw eigen batchbestanden zou kunnen gebruiken:

1 Tussen de haakjes van de FOR...DO-lus wordt rechtstreeks de inhoud van de omgevingsvariabele PATH gebruikt. Omdat de afzonderlijke directories door puntkomma's worden gescheiden, krijgt de variabele %%a achtereenvolgens steeds een directory als waarde. Met behulp van de opdracht DIR wordt de inhoud van

de directory opgevraagd en omgeleid naar het bestand SHOWPATH.TMP.

2 Het dubbele omleidingsteken '>>' zorgt ervoor dat het bestand niet opnieuw wordt aangemaakt bij iedere oproep van de opdracht DIR. In plaats daarvan wordt de nieuwe uitvoer aan het einde van het dan al bestaande bestand SHOW-PATH.DAT gekoppeld. Steeds als u de uitvoer van meerdere opdrachten wilt omleiden naar één bestand, dient u dit dubbele omleidingsteken te gebruiken. Vanzelfsprekend dient u dan wel eerst een oude versie van het bestand te wissen. Met behulp van de voorwaarde IF EXIST wordt het bestand alleen gewist, als het ook werkelijk bestaat. Zou u die voorwaarde achterwege laten, dan werd het niet-bestaande bestand natuurlijk ook niet gewist, maar het batchbestand zou in dat geval worden afgebroken met de foutmelding File not found .

3 Na de uitvoer van de gegevens naar het tijdelijke bestand worden met behulp van de opdracht FIND alle ongewenste gegevens eruit gefilterd: de bijzondere tekens '.' en '..' waarmee directories worden aangeduid en de bijzondere weergave van FIND met koppelstreepjes.

4 Om het aantal bestanden te bepalen gebruikt het batchbestand twee trucs. In de eerste plaats worden met de opdracht FIND /C "." alle regels geteld waarin een punt voorkomt. Dat zijn alle regels met een datum- en tijdsaanduiding. De uitvoer van die opdracht ziet er ongeveer als volgt uit:

```
--------- SHOWPATH.TMP: 345
```

Een speciale escape-reeks voor de driver ANSI.SYS, die wordt aangemaakt met de opdracht PROMPT, zorgt ervoor dat de cursor vervolgens drie regels omhoog wordt geplaatst. De ECHO-opdracht schrijft dan precies over die uitvoer heen, maar laat het gevonden getal staan. De witregel is daarbij van belang, u mag hem niet weglaten of onderdrukken met ECHO OFF, want dan wordt de escape-reeks niet weergegeven.

5 Steeds als u de prompt wijzigt, dient u aan het einde van het batchbestand de oude prompt weer te herstellen. Daarom wordt die in het begin van het batchbestand tijdelijk opgeslagen in de omgevingsvariabele OLDPROMPT. Uiteindelijk kan die tijdelijke variabele weer worden gewist.

8.9 Bestanden uitsluiten bij een kopieeractie

Veronderstel dat u werkt met WORD. U wilt alle gegevens uit de directory C:\WORD\DOCUMENT kopiëren, met uitzondering van de backup-bestanden. Dus u wilt tekstbestanden, grafische bestanden, tekstfragmenten, enzovoort kopiëren, tenzij ze de extensie .BAK of .SIK hebben. Dat is nog al wat werk. Het batchbestand COPYNOT.BAT neemt dat van u over.

Uitwerking

Het bestand ziet er als volgt uit:

```
@ECHO OFF
REM COPYNOT.BAT - bestanden uitsluiten bij een kopieeractie
IF "%1" == "" GOTO Syntaxis
IF "%2" == "" GOTO Syntaxis
IF NOT "%3" == "" GOTO Syntaxis
IF NOT EXIST %1 GOTO Existnot
ATTRIB +A *.* >NUL
ATTRIB -A %1 >NUL
XCOPY *.* %2 /A
IF ERRORLEVEL 1 GOTO Einde
ECHO.
ECHO Alle bestanden met uitzondering van %1
ECHO zijn gekopieerd naar drive/directory %2
GOTO Einde
:Existnot
ECHO.
ECHO De bestanden die u hebt opgegeven, komen
ECHO niet voor in deze directory
GOTO Einde
:Syntaxis
ECHO.
ECHO COPYNOT.BAT kopieert alle bestanden met uitzondering van
ECHO de bestanden die u als parameter meegeeft, naar een
ECHO drive/directory naar keuze.
ECHO Syntaxis: COPYNOT BESTAND.EXT DRIVE/DIRECTORY
ECHO.
ECHO wildcards zijn toegestaan
:Einde
@ECHO ON
```

Met dit batchbestand wordt de hele procedure teruggebracht tot één simpele opdracht:

```
COPYNOT *.BAK A:
```

Op zich is dit geen nieuwe verworvenheid van MS-DOS 5.0. Al vanaf versie 3.2 beschikt DOS, met uitzondering van de apestaart, over de functies die u voor dit batchbestand nodig hebt. Toen werd namelijk de opdracht XCOPY geïntroduceerd die een centrale rol speelt in het batchbestand.

XCOPY maakt het mogelijk alle bestanden te selecteren voor een kopieeractie waarvan het *Archive attribute* is gezet. Dat attribuut wordt bij verstek door MS-DOS gezet als u een bestand aanmaakt of wijzigt. Programma's als XCOPY en BACKUP zetten het attribuut weer op nul en geven daarmee aan dat van het bestand een kopie is gemaakt.

De truc is dus in dit geval het archiveringsattribuut zo te zetten dat alle bestanden worden gekopieerd, uitgezonderd die met de extensie .BAK. De opdracht ATTRIB +A *.* > NUL zorgt voor het eerste. Als die opdracht is uitgevoerd, lijkt het alsof alle bestanden recentelijk zijn bewerkt. De erop volgende opdracht ATTRIB -A %1 > NUL wist het attribuut van alle bestanden die overeenkomen met de variabele %1. De parameter /A bij de opdracht XCOPY zorgt er ten slotte voor dat alleen de bestanden met een gezet attribuut worden gekopieerd.

8.10 Bestanden uitsluiten bij de uitvoering van DOS-opdrachten

Het batchbestand COPYNOT werkt prima bij het kopiëren van bestanden. Wat te doen als u op dezelfde selectieve manier bestanden wilt wissen? Een batchbestand DELNOT.BAT schrijven? Het zal er voor negentig procent net zo uitzien als COPYNOT.BAT en dus kunt u beter meteen een batchbestand aanmaken waarin variabelen de plaats innemen van DOS-opdrachten.

Uitwerking

Het onderstaande batchbestand NOT.BAT is bedoeld om groepen bestanden uit te sluiten bij de uitvoering van DOS-opdrachten.

```
@ECHO OFF
REM *****************************************************
REM * NOT.BAT - bestanden uitsluiten bij DOS-opdrachten *
REM *****************************************************
IF "%1" == "" GOTO Syntaxis
IF "%2" == "" GOTO Syntaxis
IF NOT EXIST %2 GOTO Existnot
ATTRIB +H %2 >NUL
%1 *.* %3 %4 %5 %6 %7 %8 %9
ATTRIB -H %2 >NUL
GOTO Einde
:Existnot
IF NOT EXIST %2*.* GOTO Dirtest
CALL NOT.BAT %1 %2*.* %3
GOTO Einde
:Dirtest
IF NOT EXIST %2\*.* GOTO Isnot
CALL NOT.BAT %1 %2\*.* %3
GOTO Einde
:Isnot
ECHO.
ECHO *************************************
ECHO * De bestanden die u hebt opgegeven *
ECHO * komen niet in deze directory voor *
ECHO *************************************
GOTO Einde
:Syntaxis
CLS
ECHO.
ECHO *******************************************************
ECHO *  NOT.BAT voert een DOS-opdracht uit, maar sluit   *
ECHO *  bestanden die u opgeeft, daarvan uit               *
ECHO *  ----------------------------------------------     *
ECHO *  Syntaxis:  NOT OPDRACHT BESTAND.EXT PARAMETER      *
ECHO *  OPDRACHT:  DOS-opdracht, bijvoorbeeld COPY of DEL  *
ECHO *  BESTAND:   bestandsnaam, wildcards toegestaan      *
ECHO *             bijvoorbeeld TEKST.BAK of *.SIK         *
ECHO *  PARAMETER: opties en parameters bij DOS-opdrachten *
ECHO *             bijvoorbeeld A: of /P                   *
ECHO *******************************************************
:Einde
@ECHO ON
```

Het batchbestand maakt gebruik van een nieuwe mogelijkheid van MS-DOS 5.0. Tot en met versie 4.0 mochten met de opdracht ATTRIB alleen de attributen *Archive* en *Read only* worden gemanipuleerd. MS-DOS 5.0 staat echter ook toe de attributen *Hidden* en *System* te wijzigen. Het attribuut Hidden zorgt ervoor dat de meeste DOS-opdrachten geen greep meer hebben op het bestand waarvoor het is gezet. Dat is precies wat u nodig hebt voor dit batchbestand.

Het batchbestand werkt als volgt: Eerst wordt het attribuut Hidden van alle geselecteerde bestanden (de tweede variabele) gezet met behulp van de opdracht ATTRIB +H %2 >NUL. De omleiding naar NUL verhindert dat de gebruikelijke meldingen van de opdracht op het scherm worden weergegeven. Aansluitend wordt de opgegeven DOS-opdracht (de eerste variabele) uitgevoerd voor alle overige bestanden (*.*). Ten slotte wordt het attribuut Hidden van de geselecteerde groep weer gewist.

De opdracht uit de vorige paragraaf, waarmee u alle bestanden, uitgezonderd die met de extensie .BAK, naar A: kopieerde, kunt u nu dus op de volgende manier met NOT.BAT uitvoeren:

```
NOT COPY *.BAK A:
```

De opdracht laat zich zo ongeveer lezen als 'kopieer *.* maar niet *.BAK'.

Vanzelfsprekend kunt u NOT.BAT ook gebruiken samen met de opdracht DEL (wees wel voorzichtig). Aan het begin van de vorige paragraaf namen we een directory als uitgangspunt die vol stond met tekstbestanden (.TXT) en bijbehorende bestanden. Veronderstel dat u om ruimte te maken al die bijbehorende bestanden wilt wissen. Met dit batchbestand is dat geen enkel probleem. U typt de volgende opdracht:

```
NOT DEL *.TXT
```

Tijdens de uitvoering van het batchbestand zult u de controlevraag 'Are you sure (Y/N)?' moeten beantwoorden. Die wordt namelijk altijd gesteld door DOS als u de opdracht DEL *.* geeft, die in het batchbestand voorkomt. Omdat u de tekstbestanden zelf al hebt beschermd, kunt u rustig 'Y' typen.

Het batchbestand in zijn huidige vorm heeft één belangrijk nadeel: als zich in de directory al bestanden bevinden waarvan het attribuut Hidden is gezet, stuurt u die status aan het eind in de war. Zou u alle bestanden in een directory willen wissen met uitzondering van een paar bestanden met de extensie .EXE, dan kunt u het batchbestand gebruiken. Bevinden er zich in de directory ook nog twee bestanden met de extensie .EXE, waarvan het attribuut Hidden is gezet, dan zullen die uiter-

aard ook niet worden gewist, maar wel aan het eind weer zichtbaar worden gemaakt. Het aanpassen van het huidige batchbestand leidt tot een uitbreiding van een paar pagina's. We stellen daarom een eenvoudiger oplossing voor die in de praktijk goed voldoet: het probleem komt zelden voor.

We wijzigen het batchbestand zo, dat de bestanden waarvan aanvankelijk het attribuut Hidden is gezet, eerst worden getoond op uw scherm. U kunt ze dan na uitvoering van het batchbestand weer verbergen. Voeg aan het begin van het batchbestand tussen de beide regels:

```
IF NOT EXIST %2 GOTO Existnot
ATTRIB +H %2 >NUL
```

de volgende regels toe:

```
CLS
ECHO.
ECHO ****************************************************************
ECHO *                    LET OP!                                   *
ECHO * Van de bestanden die moeten worden uitgesloten bij de        *
ECHO * uitvoering van de opdracht, is een aantal wellicht verborgen, *
ECHO * omdat het attribuut HIDDEN is gezet. Dit geldt voor de       *
ECHO * bestanden die na deze mededeling op uw scherm verschijnen.   *
ECHO * Na beëindiging van dit batchbestand zult zelf het attribuut  *
ECHO * HIDDEN van deze bestanden opnieuw moeten zetten!             *
ECHO * Druk op een toets...                                         *
ECHO ****************************************************************
ECHO.
PAUSE >NUL
DIR %2 /A:H
ECHO.
ECHO ****************************************************************
ECHO * Noteer indien nodig de namen van de bovenstaande bestanden   *
ECHO * Zet het attribuut HIDDEN met behulp van de opdracht ATTRIB   *
ECHO * na beëindiging van het batchbestand opnieuw.                 *
ECHO * Druk op een toets...                                         *
ECHO * AFBREKEN met Ctrl-C                                          *
ECHO ****************************************************************
ECHO.
PAUSE >NUL
```

Dit stukje programma stelt u in staat de namen van eventuele verborgen bestanden te noteren en achteraf het attribuut Hidden opnieuw te zetten met de opdracht ATTRIB. Stel dat er een verborgen bestand OPDRACHT.EXE wordt getoond, terwijl u het batchbestand hebt opgeroepen met 'NOT DEL *.EXE'. Het verborgen bestand wordt in het begin getoond en na beëindiging van het batchbestand geeft u de volgende opdracht om het attribuut weer te zetten:

```
ATTRIB OPDRACHT.EXE +H
```

9 Beveiliging

Dit hoofdstuk is bedoeld voor diegenen onder u die samen met andere gebruikers op één PC werken. Verkeert u in de gelukkige omstandigheid dat u de enige bent die achter het toestel zit en van gegevensbestanden gebruik maakt, dan is dit hoofdstuk voor u waarschijnlijk minder interessant.

9.1 Registratie van gebruik

Als u het gebruik van uw PC door anderen niet wilt of kunt verhinderen, maar toch wilt weten wat er zich tijdens uw aanwezigheid afspeelt, is het handig als de PC de momenten bijhoudt waarop er is gewerkt.

Uitwerking

De PC dient de datum en het tijdstip waarop hij wordt gebruikt vast te leggen. Beide gegevens moeten worden omgeleid naar een bestand dat u dan kunt raadplegen. We hebben gekozen voor een klein bestand dat alleen het laatste gebruik opslaat. Omdat u om dat bestand te kunnen raadplegen, eerst zelf uw PC moet inschakelen, en dan dus die gegevens zult aantreffen, zijn er twee dergelijke bestanden nodig. Het eerste daarvan wordt onmiddellijk na het inschakelen van de PC beveiligd; pas in tweede instantie worden dan de huidige gegevens opgeslagen.

Om te voorkomen dat andere gebruikers uw registratie ontdekken en in de war sturen, mogen er geen meldingen naar het scherm worden gestuurd. Daarnaast maakt het programma gebruik van de opdrachten DATE en TIME. Die moeten altijd worden afgesloten met Enter. Door dat te automatiseren vestigen we geen aandacht op het hele proces. Alles met elkaar hebt u het volgende nodig:

1 Een directory C:\TEMP waarin u de datum en het tijdstip opslaat. Vanzelfsprekend kunt u ook een andere directory opgeven.

2 Een directory C:\BAT met een bestand ENTER voor het omleiden van de opdrachten DATE en TIME. Ook hier kunt u een andere directory opgeven.

3 Het bestand ENTER, dat u als volgt aanmaakt. Met de volgende opdracht opent u een nieuw bestand:

```
COPY CON C:\BAT\ENTER
```

De inhoud daarvan bestaat slechts uit één handeling: u drukt één keer op Enter. Daarna slaat u het bestand op met Ctrl-Z.

4 Drie nieuwe regels in het bestand AUTOEXEC.BAT. Ze kopiëren de inhoud van het bestand TIJD naar het bestand TIJD.OLD en schrijven vervolgens de datum en het tijdstip naar het bestand TIJD. Het zijn de volgende regels:

```
@COPY >NUL C:\TEMP\TIJD C:\TEMP\TIJD.OLD
@TIME >C:\TEMP\TIJD <C:\BAT\ENTER
@DATE >>C:\TEMP\TIJD <C:\BAT\ENTER
```

Tip: Vindt u deze drie regels nog te opvallend in het bestand AUTOEXEC.BAT, dan slaat u ze op in een apart batchbestandje, bijvoorbeeld onder de naam CHKD$K.BAT. U roept dat bestand dan op met de opdracht CALL. Deze naam lijkt zoveel op die van het programma CHKDSK, dat het veel gebruikers niet op zal vallen dat het om iets heel anders gaat.

Roept u het bestand AUTOEXEC.BAT voor de eerste keer op in zijn nieuwe vorm, dan bestaat er uiteraard nog geen bestand TIJD. Dat levert de foutmelding 'File not found' op. Om ervoor te zorgen dat niemand door die melding wantrouwend wordt, dient u het bestand AUTOEXEC.BAT dus één keer helemaal te doorlopen.

Na deze aanpassing kunt u steeds controleren wanneer de PC voor het laatst is ingeschakeld, met de opdracht:

```
TYPE C:\TEMP\TIJD.OLD
```

Toelichting

De apestaart voor de drie nieuwe regels in het bestand AUTOEXEC.BAT zorgt ervoor dat de regels niet op het scherm verschijnen. De melding '1 file(s) copied' zou echter nog steeds op uw scherm getoond worden na de COPY-opdracht en daarom wordt de uitvoer van die opdracht omgeleid naar het randapparaat NUL, dat niets doet: het slokt gegevens eenvoudigweg op en laat ze nooit weer te voorschijn komen.

In de tweede en de derde regel van het stukje leiden we de uitvoer van de opdrachten DATE en TIME om naar het bestand TIJD. Uit het bestand ENTER halen we de

toetsaanslag op Enter die nodig is om beide gegevens te bevestigen. We bootsen met dit bestand dus een toetsaanslag na.

De eerste omleiding gebeurt met het omleidingsteken '>'. Om te voorkomen dat de tweede omleiding over de eerste heen schrijft, gebruiken we daarna het teken '>>', dat ervoor zorgt dat de nieuwe gegevens aan de oude worden vastgekoppeld.

9.2 Handelingen registreren

De aanpassing van de vorige paragraaf vertelt u hoogstens dàt er iemand aan uw PC heeft gezeten; het is nogal frustrerend om niet te weten wàt hij heeft gedaan. Als u uw PC korte tijd in de steek laat zonder hem uit te zetten, kunt u daar snel achter komen met behulp van het programma DOSKEY.

Uitwerking

Zorg ervoor dat DOSKEY op uw systeem actief is. Druk na terugkeer een paar maal op cursor omhoog en achter de systeemprompt verschijnen de laatst ingevoerde opdrachten. Drukt u op F7, dan krijgt u een overzicht van alle opdrachten. Daaraan valt snel genoeg af te lezen of uzelf of iemand anders als laatste actief is geweest.

9.3 Een gebruikerslogboek bijhouden

Een volgende uitbreiding van de beveiliging betreft het bijhouden wie er op de PC hebben gewerkt. Daartoe maakt u een logboek. Iedere gebruiker die met de PC wil werken, dient eerst zijn naam in te voeren; daaraan worden dan datum en tijdstip gekoppeld.

Uitwerking

U hebt voor het logboek een klein programmaatje nodig in machinetaal en een paar extra regels in het bestand AUTOEXEC.BAT. Het programma in machinetaal tekent de gebruikersnaam op; met de variabelen die u in batchbestanden mag gebruiken, is dat niet mogelijk.

Start EDIT en maak het onderstaande programma INPUT.DEB aan. Leest u paragraaf 1.6 nog eens door als u niet meer precies weet hoe u programma's in machinetaal maakt.

```
A
MOV AH,08
INT 21
MOV DL,AL
MOV AH,02
INT 21
CMP AL,0D
JNZ 0100
MOV DL,0A
INT 21
MOV AL,00
MOV AH,4C
INT 21

RCX
18
Ninput.com
W
Q
```

U converteert dit bestand naar een programma in machinetaal met de debugger van MS-DOS:

```
DEBUG <INPUT.DEB
```

Het resultaat is het programma INPUT.COM in de actuele directory. Kopieer het programma naar een directory waarnaar een zoekpad verwijst, bijvoorbeeld C:\BAT.

Het programma wordt geactiveerd vanuit het bestand AUTOEXEC.BAT. Daartoe voegt u de volgende regels aan dat bestand toe:

```
@ECHO OFF
BREAK OFF
IF EXIST C:\TEMP\USER.LOG COPY >NUL C:\TEMP\USER.LOG
C:\TEMP\USER.OLD
ECHO Voer uw naam in
INPUT >C:\TEMP\USER.LOG
DATE <C:\BAT\ENTER >>C:\TEMP\USER.LOG
TIME <C:\BAT\ENTER >>C:\TEMP\USER.LOG
BREAK ON
@ECHO ON
```

De eerstvolgende keer dat u het bestand AUTOEXEC.BAT activeert, verschijnt op uw scherm de opdracht:

```
Voer uw naam in
```

U kunt nu uw naam typen, die overigens niet op het scherm verschijnt. Dat levert verder geen problemen op, maar maakt het wel mogelijk het programma ook te gebruiken voor het invoeren van een wachtwoord. Aansluitend worden naam, datum en tijdstip in het bestand USER.LOG in de directory C:\TEMP opgeslagen.

Toelichting

In de eerste plaats wordt gecontroleerd of het bestand USER.LOG al bestaat. Als dat het geval is, wordt er onder de naam USER.OLD een kopie van gemaakt. Daarna verschijnt de opdracht een naam in te voeren, die met behulp van het machinetaal-programma INPUT.COM wordt opgeslagen.

Dat programma leest tekens van het actuele invoerapparaat (in dit geval het toetsenbord) totdat op Enter wordt gedrukt. Die tekens worden uitgevoerd naar het actuele uitvoerapparaat, in dit geval het bestand USER.LOG.

Met behulp van de beide omleidingstekens en het kleine bestandje ENTER, dat we in de vorige paragraaf hebben besproken, koppelen we er ten slotte nog de datum en het tijdstip aan.

9.4 Registratie van gebruiksduur

In deze paragraaf gaan we wat verder op de ingeslagen weg. We laten u zien hoe u niet alleen kunt vastleggen wanneer de PC in gebruik is genomen, maar ook hoe lang erop is gewerkt.

Uitwerking

De meest eenvoudige methode om vast te leggen hoe lang de PC in gebruik is geweest, maakt gebruik van het batchbestand EINDE.BAT. U roept de gebruiker op dit batchbestand te activeren voordat hij de computer uitzet. Het batchbestand kan er als volgt uitzien:

```
@ECHO OFF
ECHO >>C:\TEMP\USER.LOG uitgeschakeld om
DATE <C:\BAT\ENTER >>C:\TEMP\USER.LOG
TIME <C:\BAT\ENTER >>C:\TEMP\USER.LOG
ECHO ON
```

Het batchbestand vertelt u echter niets als een gebruiker het niet activeert. Bovendien wilt u misschien helemaal niet dat een gebruiker weet dat u zijn werktijd registreert. Dat laatste kunt u verhelpen als u de gebruiker bijvoorbeeld opdraagt voor het uitschakelen een programma voor het parkeren van de koppen van de harde schijf te activeren: PARK.COM. Het echte programma PARK.COM hernoemt u tot bijvoorbeeld _PARK.COM en u maakt een batchbestand PARK.BAT aan, dat er als volgt uitziet:

```
@ECHO OFF
ECHO >>USER.LOG uitgeschakeld om
DATE <C:\BAT\ENTER >>C:\TEMP\USER.LOG
TIME <C:\BAT\ENTER >>C:\TEMP\USER.LOG
_PARK
ECHO ON
```

Er zijn twee redenen waarom u het oorspronkelijke programma even een andere naam moet geven. In de eerste plaats zou het batchbestand leiden tot een eindeloze lus, omdat de voorlaatste opdracht dan opnieuw het batchbestand zou oproepen; in de tweede plaats zou niet het batchbestand worden opgeroepen, maar het oorspronkelijke programma als toevallig de directory C:\DOS actief was.

U kunt nu de gebruiker via bijvoorbeeld het bestand AUTOEXEC.BAT een opdracht geven om bij het afsluiten om veiligheidsredenen beslist de koppen van de harde schijf te parkeren. De meeste gebruikers zullen zonder argwaan aan die oproep gehoor geven.

Toelichting

Het is van belang het oorspronkelijke programma PARK te hernoemen tot _PARK, zodat de gebruiker de eigenlijke naam typt in de waan de koppen te parkeren; intussen echter heeft het batchbestand razendsnel datum en tijd aan het bestand USER.LOG gekoppeld.

9.5 Beveiligen met een wachtwoord

Als u even uw computer in de steek laat, maar u wilt het apparaat niet uitzetten, kunt u met behulp van een klein programma in BASIC verder werken koppelen aan een wachtwoord. Andere gebruikers kunnen dan tijdens uw afwezigheid niet met de computer werken.

Uitwerking

Maak het volgende programma WW.BAS met de editor van QBASIC:

```
REM WW.BAS
WW$ = "msdos"
REM
INVOER:
CLS
FOR i = 1 TO LEN(WW$)
  N$ = MID$(WW$,i,1)
  GOSUB TEKEN
  IF N$ <> A$ THEN
    BEEP
    GOTO INVOER
  END IF
NEXT i
SYSTEM
REM
TEKEN:
A$ = INKEY$
IF A$ = "" THEN GOTO TEKEN
PRINT ".";
RETURN
```

Sla het programma op onder de naam WW.BAS (WachtWoord.BAS) in de directory C:\BAT. Als u het start, wordt het scherm gewist en wacht het programma op de invoer van de juiste toetsaanslagen, in dit geval ‘msdos’. Iedere juiste toetsaanslag tovert een punt op het scherm, de tekens zelf worden niet weergegeven, zodat niemand het wachtwoord op het scherm kan lezen.

Als het hele wachtwoord juist is ingevoerd, wordt het programma beëindigd. Slaat u een foute toets aan, dan klinkt een waarschuwingssignaal en moet het hele wachtwoord van voren af aan weer worden opgebouwd. U start het programma het snel-

ste met een klein batchbestand WW.BAT:

```
@ECHO OFF
@QBASIC /RUN C:\BAT\WW.BAS
@ECHO ON
```

U roept via dit batchbestand het programma op met de opdracht 'WW'. U kunt het programma niet afbreken met Ctrl-C, omdat die toetsencombinatie in QBASIC het teken met ASCII-code 3 genereert en niet onmiddellijk tot afbreken leidt. Om het programma af te breken drukt u op Ctrl-Break.

Wilt u helemaal verhinderen dat een gebruiker het programma afbreekt, dan compileert u het programma met de compiler van QUICKBASIC. Dat schakelt ook de toetsencombinatie Ctrl-Break uit.

9.6 Gegevens definitief wissen

Een van de belangrijke verbeteringen van MS-DOS 5.0 is de wijziging van de opdrachten DEL en FORMAT in die zin, dat ze gegevens niet onmiddellijk zo wissen dat u de actie niet kunt herstellen. Een vervelende bijkomstigheid daarvan is dat u niet zonder meer uw computer uit handen kunt geven bij bijvoorbeeld verkoop of reparatie. Zou u in dat geval de harde schijf (na het maken van een backup!) formatteren, dan kan een volgende gebruiker zonder problemen de bestanden weer leesbaar maken. Het programma VUL.BAS, dat we in deze paragraaf bespreken, voorkomt dat.

Uitwerking

Onder MS-DOS 5.0 worden bestanden na een opdracht DEL of FORMAT alleen gemarkeerd als gewist. De eerste letter van de bestandsnaam wordt onleesbaar gemaakt en de geheugenruimte die het bestand op de schijf inneemt, wordt vrijgegeven. Het bestand zelf echter bestaat nog steeds. Het kan tijden duren voordat nieuwe gegevens naar de vrijgekomen ruimte worden geschreven.

Het onderstaande programma VUL.BAS schrijft zoveel gegevens naar de harde schijf, dat alle geheugenruimte verbruikt wordt en dus geen oude gegevens meer leesbaar zijn.

```
REM VUL.BAS - schrijft de harde schijf vol met een teken
OPEN "C:\VUL.DAT" FOR OUTPUT AS #1
```

```
    N$ = STRING$(512,"*")
    CLS
    VOL = 0
    ON ERROR GOTO Fout
    WHILE VOL = 0
      PRINT #1, N$
      PRINT ".";
    WEND
  CLOSE #1
  PRINT
  BEEP
  PRINT "Bestand aangemaakt..."
  WHILE INKEY$ = "" : WEND
  KILL "C:\VUL.DAT"
  PRINT "Bestand gewist"
  END

  Fout:
  IF ERR = 61 THEN
    VOL = 1
    RESUME NEXT
  ELSE
    PRINT "Fout: "; ERR ; "opgetreden; het programma wordt beëindigd."
    END
  END IF
```

Nadat u het programma start, worden steeds 512 tekens naar een bestand VOL.DAT geschreven. Dit gaat net zo lang door tot de harde schijf vol is. Op het scherm wordt bij ieder lus een punt weergegeven, zodat u het proces min of meer kunt volgen. Alles met elkaar komen er heel wat punten op het scherm te staan...

Na beëindiging wordt VOL.DAT weer gewist.

Tip: Er bestaan uiteraard ook speciale programma's voor dit werk. Een van de bekendste is het programma WIPEINFO van de Norton Utilities 5.0.

9.7 De computer starten met een wachtwoord

Een volgende stap op het gebied van beveiliging is de toegang tot de computer te koppelen aan het juist invoeren van een wachtwoord. Dat kan met behulp van het programma WW.BAS van paragraaf 9.5.

Uitwerking

Voeg aan het bestand AUTOEXEC.BAT een regel toe die het BASIC-programma WW.BAS oproept. Dat heeft eigenlijk alleen zin als u het programma eerst compileert, zodat een gebruiker het niet kan afbreken met Ctrl-Break. Ook dan werkt de beveiliging natuurlijk nog niet voor 100 procent, omdat iemand met een systeemdiskette het wachtwoord omzeilt door het systeem te starten vanaf drive A:.

Alleen in het ongecompliceerde geval dat u de enige eigenlijke gebruiker van de computer bent, heeft deze procedure zin. Zijn er ook andere gebruikers, dan zult u hen het wachtwoord moeten geven. Het is een bekend gegeven dat als twee een geheim kennen, het geheim niet meer bestaat. De volgende paragraaf helpt u verder gegevens te beschermen als meerdere gebruikers op dezelfde computer moeten werken.

9.8 Bestanden en directories beveiligen

Als meerdere gebruikers een PC delen, heeft het soms zin hen tegen elkaar in bescherming te nemen. Er zijn diverse manieren waarop u dat kunt doen. We zullen ze hieronder de revue laten passeren.

Uitwerking

De meest eenvoudige en tevens meest veilige methode is het consequent opslaan van gegevensbestanden op diskettes, die u achter slot en grendel bewaart. Als het echter om grote gegevensbestanden gaat die u steeds nodig hebt, is dit niet zo praktisch.

Een tweede methode is het verbergen van de bestanden. MS-DOS zelf maakt van die mogelijkheid gebruik om twee belangrijke systeembestanden op te slaan: IBMBIO.COM EN IBMDOS.COM. Met de gebruikelijke opdrachten DEL, COPY, DIR en dergelijke kunt u die bestanden niet bereiken. Wel kunt u de bestandsattributen wijzigen met de DOS-opdracht ATTRIB, zodat eigenlijk geldt dat de beveiliging alleen werkt voor niet-ingewijden. Ook de DOS Shell toont eventueel bestanden waarvan het attribuut hidden is gezet.

Bestanden verbergen met de opdracht ATTRIB

De eerste manier om bestanden te verbergen werkt met de opdracht ATTRIB. Voer

achter de opdracht de naam van het bestand in, eventueel voorzien van zoekpad en voeg de optie +H (Hide) toe. Neem om dit uit te proberen een lege, geformatteerde diskette, kopieer het bestand COMMAND.COM naar die diskette en verberg het bestand vervolgens met de opdracht:

```
ATTRIB A:\COMMAND.COM +H
```

Geef de opdracht DIR A: om na te gaan of het gelukt is. Het systeem zal melden dat er geen bestand is gevonden. Zo lang COMMAND.COM het enige bestand op de diskette is, wordt de illusie dat er helemaal geen bestand is nog versterkt, omdat DOS, evenals bij lege diskettes, geen melding maakt van het aantal bytes dat nog vrij is. Pas als u een gewoon, leesbaar bestand naar de diskette kopieert, zal het een oplettende gebruiker opvallen, dat er bytes 'verdwenen' zijn.

Het verborgen bestand kan niet meer worden gewist of gekopieerd. Om de oefen-diskette weer schoon op te kunnen bergen zult u COMMAND.COM eerst weer zichtbaar moeten maken. Dat kan met dezelfde opdracht ATTRIB, nu met de optie -H:

```
ATTRIB A:\COMMAND.COM -H
```

Directories verbergen

Als u veel bestanden wilt verbergen, is het wat omstandig voor al die bestanden het attribuut Hidden te zetten. Het is dan eenvoudiger ze allemaal op te slaan in een verborgen directory. Ook directories kunt u verbergen met de opdracht ATTRIB. In tegenstelling tot verborgen bestanden kunt u echter de bestanden van verborgen directories wel manipuleren met DOS-opdrachten als DEL en dergelijke, zo lang u de naam maar weet.

Slaat u bijvoorbeeld gewoonlijk al uw eigen bestanden op in een directory C:\HENK, dan voert u de volgende opdracht in om die directory onzichtbaar te maken:

```
ATTRIB C:\HENK +H
```

Toch kunt u gewoon met de bestanden werken. De volgende opdracht activeert de directory:

```
CD C:\HENK
```

Een daarop volgende opdracht DIR geeft alle bestanden in die directory gewoon weer op uw scherm.

Bestanden verbergen vanuit de DOS Shell

Vanuit de DOS Shell kunt u ook heel eenvoudig bestanden verbergen. Daartoe markeert u een bestand en kiest in het menu File de optie Change Attributes.... Er wordt een dialoogvenster geopend waarin u kunt kiezen welke van de vier attributen u wilt wijzigen.

Om het proces weer ongedaan te maken moet u de verborgen bestanden natuurlijk eerst in de DOS Shell kunnen bekijken en markeren. U opent daartoe het menu Options en kiest File Display Options.... In het dialoogvenster kiest u Display hidden/system files. Vervolgens worden de verborgen bestanden weergegeven en kunt u de attributen ervan wijzigen. In de DOS Shell kunt u geen directories onzichtbaar maken.

9.9 Bestanden en directories met een bepaalde naam verbergen

De opdracht CHKDSK trekt zich van de beveiliging die we in de vorige paragraaf hebben besproken in zoverre niets aan dat de bestandsnamen gewoon worden weergegeven. Een handige gebruiker kan met behulp van die opdracht achter de namen van bestanden en directories komen, er vervolgens de attributen van wijzigen, enzovoort. Er is echter een methode om die gebruiker nog wat zand in de ogen te strooien door niet-leesbare tekens te gebruiken in een bestandsnaam.

Uitwerking

De eenvoudigste manier om niet-leesbare tekens te genereren is de toetsencombinatie Alt-255 (num): U houdt de Alt-toets ingedrukt en drukt vervolgens op het numerieke toetsenbord de toetsen 2, 5 en 5 in. Geeft u dit teken als eerste letter van een bestandsnaam, dan beschouwt DOS het als een spatie. Ga als volgt te werk om zo'n bestandsnaam of directory-naam te maken.

1 Typ Alt-255. Op uw scherm verspringt de cursor één positie naar rechts.

2 Typ de bestandsnaam of directory-naam zoals gebruikelijk. Houdt u daarbij wel

aan de regel dat een naam voor de punt maximaal acht tekens, inclusief Alt-255, lang mag zijn.

Een directory met een dergelijk bestand ziet er als volgt uit:

```
 Volume in drive C has no label
 Volume Serial Number is 1435-078B
 Directory of C:\

AUTOEXEC BAT       349 24-10-91   11:15
COMMAND  COM     47845 09-04-91    5:00
CONFIG   SYS       370 28-09-91   22:14
TEST                23 25-10-91   00:01
 TEST               23 25-10-91   00:02
```

Het tweede bestand TEST wijkt qua plaatsing nogal af van de overige. Dat valt nogal op in een bestandenlijst. Vandaar dat we u nog een variatie op de truc bieden. Wat vindt u van het volgende overzicht:

```
 Volume in drive C has no label
 Volume Serial Number is 1435-078B
 Directory of C:\

AUTOEXEC BAT       349 24-10-91   11:15
COMMAND  COM     47845 09-04-91    5:00
CONFIG   SYS       370 28-09-91   22:14
TEST                23 25-10-91   00:01
TEST                23 25-10-91   00:02
```

Wellicht voor het eerst ziet u hier twee bestanden met dezelfde naam, iets wat onder MS-DOS helemaal niet mogelijk is. Het is geen drukfout, maar een toepassing van Alt-255. Het tweede bestand heeft als extensie de toetsaanslag Alt-255 en dus beschouwt DOS het als een volstrekt unieke naam.

Met behulp van deze truc kunt u veel gebruikers de toegang tot bestanden ontzeggen: u voegt aan het eind van de naam of als extensie Alt-255 toe en de argeloze gebruiker zal zich erover verwonderen dat hij geen greep heeft op het bestand, ook al voert hij de bestandsnaam nog zo zorgvuldig in. Iedere keer dat hij de opdracht

```
COPY TEST A:
```

geeft, meldt het systeem ‘File not found’.

Voor het geval u zelf mocht zijn vergeten hoe het u het had gedaan, het volgende: MS-DOS kent wildcards; vul de naam aan met sterretjes of vraagtekens en DOS reageert weer als vanouds. Ook de gevorderde gebruiker zal daarvan overigens gebruik maken.

9.10 Bijna perfecte beveiliging

Nadat u de truc hebt toegepast om CHKDSK te slim af te zijn, blijft het probleem dat de DOS Shell op verzoek alle verborgen bestanden laat zien. Met de optie View file contents uit het menu File kunt u zelfs de inhoud van dergelijke bestanden bekijken. U kunt echter de naam van een directory zo wijzigen, dat ook dat niet meer lukt. Daarvoor dient u dan wel te beschikken over een programma als PC TOOLS of de Norton Utilities waarmee u de inhoud van programmabestanden kunt wijzigen.

Uitwerking

Maak vanuit de hoofddirectory van de harde schijf een directory NULL aan:

```
MD C:\NULL
```

Kopieer alle bestanden die u wilt beschutten voor vreemde blikken naar die directory. Start vervolgens PC TOOLS of de Norton Utilities en haal een overzicht van de harde schijf naar het scherm. In PC TOOLS kiest u View/Edit Disk in het menu Disk. Open met F6 het dialoogvenster waarin u een sector kunt kiezen en kies de sector First ROOT DIR sector. Zoek vervolgens de directory NULL. U vindt de volgende regel:

```
                      ↓
0352(0160) 4E 55 4C 4C 20 20 20 20 20 20 20 10 00 00 00 00 NULL
```

Druk op F7 om de inhoud te wijzigen en vervang de waarde '4C' onder de pijl hierboven door '20'. Druk op F5 om de wijziging op te slaan en bevestig de controlevraag. Het lijkt allemaal wat eng, maar als u te werk gaat zoals we hier beschrijven, kan er niets fout gaan. De directory heet nu kennelijk C:\NUL, wat de opdracht DIR en de DOS Shell beide laten zien. Wat u ook doet, u kunt niets meer met de directory. De toegang wordt u geweigerd. Pas nadat u in PC TOOLS de oorspronkelijke toestand hebt hersteld ('20' wordt weer '4C'), hebt u weer toegang tot de directory.

9.11 Bestandsresten wissen

Een diskette waarop meerdere malen bestanden geschreven en gewist zijn, bevat naast alle zichtbare en verborgen bestanden ook nog resten van bestanden. U kunt die resten wissen met behulp van speciaal voor dit doel geschreven software van bijvoorbeeld Norton of met een klein eigen programma CLEANF.PAS. Zo kunt u voorkomen dat buitenstaanders aan wie u de diskette voor een bepaald doel geeft, geen inzicht krijgen in informatie die niet voor hen bestemd is.

Uitwerking

Het makkelijkst schoont u een diskette op als u beschikt over de Norton Utilities.

De Norton Utilities tot versie 4.5

Dit pakket bevat onder andere de programma's WIPEDISK en WIPEFILE. Het eerste van de twee schrijft een willekeurig teken naar alle geheugenplaatsen van de diskette, die daardoor volstrekt onleesbaar wordt. Het tweede wist een bestand op een soortgelijke manier definitief.

Gebruikt u WIPEDISK zonder parameters, dan wordt over alle gegevens van de hele schijf inclusief de bootsector geschreven. Daarmee gaan ook bestanden met het attribuut *Read only* verloren. Wilt u alleen vrije plaatsen tussen bestanden en lege sectoren schoonmaken, dan gebruikt u de optie /E. Het programma schrijft dan een nul naar die sectoren, tenzij u met de parameter /Vn zelf een ander teken opgeeft.

De Norton Utilities versie 5.0

Als u een bestand kopieert onder MS-DOS, bevat het altijd sectorresten met niet tot het bestand behorende gegevens. Vaak gaat het om gegevens over de inhoud van een directory, die dus ongewild iets meedelen over wat u op uw harde schijf hebt staan. Maar ook andere bestandsresten kunnen worden meegekopieerd.

In versie 5.0 van de Norton Utilities zijn de programma's WIPEDISK en WIPEFILE samengevoegd met andere utilities in het programma WIPEINFO. Een van de functies van dit programma is het wissen van deze verraderlijke sectorresten. Roep daartoe het programma op met de optie /K om de *slacks* te wissen. Voor een bestand TEST.PAS in de hoofddirectory van drive A: luidt de opdracht als volgt:

```
WIPEINFO A:\TEST.PAS /K
```

Wilt u alle bestanden op die diskette opschonen, dan typt u:

```
WIPEINFO A:\*.* /K
```

Wilt u gewiste bestanden en ongebruikte sectoren van gegevensresten ontdoen, dan voert u de opdracht in met de parameter /E. Voor de diskette in A: ziet de opdracht er zo uit:

```
WIPEINFO A: /E
```

Uw eigen schoonmaakbedrijf: CLEANF.PAS

Beschikt u niet over de Norton Utilities, dan kunt u zelf een klein programma schrijven in Turbo Pascal. Het programma CLEANF.PAS schrijft een uitroepteken naar alle bytes van sectoren vanaf de EOF-marker (End Of File) tot het sectoreinde.

Als de eerstvolgende sector (de eerstvolgende 512 bytes) weer gegevens van het bestand bevat, wordt daar niet overheen geschreven, net zo min als WIPEDISK van de Norton Utilities 4.5 dat doet. Die gegevens kunnen dan best achterhaald zijn; het feit dat de sector nog steeds voor het bestand is gereserveerd, heeft te maken met het feit dat DOS bestanden niet per sector wegschrijft, maar per cluster. Een cluster op een schijf met een capaciteit van 360 Kb bestaat uit twee sectoren.

Kopieert u echter opgeschoonde bestanden naar een net geformatteerde diskette (onder MS-DOS 5.0 met de optie /U!), dan zijn dergelijke clusterresten natuurlijk leeg. Ze worden niet meegekopieerd, maar vrijgehouden. Het opschonen van bestanden die naar een net geformatteerde schijf zijn gekopieerd, is even effectief.

Tip: De tekstverwerker WORD sluit ieder tekstbestand (met de extensie .TXT) automatisch af bij de eerstvolgende sectorgrens. Het is daarom niet nodig dergelijke bestanden op te schonen met CLEANF. Het programma zal dan steeds de melden:

```
Bestand afgesloten bij sectorgrens. Geen actie ondernomen.
```

Het onderstaande programma kunt u invoeren met de tekstverwerker van Turbo Pascal vanaf versie 5.0 en vervolgens compileren.

```
{$M 16384,0,16384}          (* Stack=16384, Heapmin=0, Heapmax=16384 *)
```

```
program CleanF;

{ Schoont de sectorresten van een opgegeven bestand op}
{ Syntaxis: CLEANF BESTAND.EXT }
uses Dos, Crt;

VAR
 Fout:Integer;
  f: file of byte;
  ff: text;
  i,j: integer;
  size:LONGINT
  rest:integer;
  bestandsnaam:string;
  teken:Byte;
BEGIN
  teken:=BYTE ('!');
  IF Paramcount<>1 THEN
  BEGIN
    Writeln ('Syntaxis: CLEANF BESTAND.EXT');
    HALT(1);                      { ongeldig aantal parameters }
  END;

  bestandsnaam:=Paramstr(1);
  {$I-}
  Assign(f, bestandsnaam);
  Reset(f);
  {$I+}
  IF IOResult<>0 THEN
  BEGIN
    Writeln ('Het bestand is niet gevonden.');
    HALT(2);
  END;
  Writeln ('Opschonen van het bestand ', bestandsnaam);
  size:=Filesize(f);
  rest:=512 - (size MOD 512);
  IF rest= 512 THEN
  BEGIN
    Writeln ('Bestand afgesloten bij sectorgrens.');
    Writeln ('Geen actie ondernomen.');
    HALT(0);
  END;
```

```
    Writeln ('Werkelijke grootte: ', size);
    Writeln ('Op te schonen rest: ', rest);
    Close(f);

    Assign(f, bestandsnaam);
    Reset(f);
    Seek(f, size);
    I:=0;
    Write ('Schrijf nu teken: ');
    WHILE i < rest DO
    BEGIN
      Write (f, teken);
      Write ('.');
      inc(i);
    END
    Close(f);
    Writeln;

    Assign(f, bestandsnaam);
    Reset(f);
    Seek(f, size);
    truncate(f);
    close(f);
    Writeln ('Opdracht voltooid.');
  END.
```

Compileer het programma nadat u het hebt geschreven. Wilt u vervolgens het bestand TEST.PAS op diskette doorgeven aan iemand anders zonder dat allerlei bestandsresten ongewild meegaan, dan formatteert u eerst een diskette met de opdracht:

```
FORMAT A: /U
```

Vervolgens kopieert u het bestand TEST.PAS naar deze diskette en start u het programma CLEANF met de opdracht:

```
CLEANF A:\TEST.PAS
```

Het programma meldt u hoe groot TEST.PAS is en hoe groot de sectorrest is die moet worden opgeschoond. Daarna wordt een uitroepteken naar alle bytes van die sectorrest geschreven, wat voor u wordt bijgehouden op het scherm.

Toelichting

De oorzaak van het hele probleem zit hem in het feit dat MS-DOS de opdracht DEL niet letterlijk uitvoert. Alleen de verwijzing naar een bestand wordt verwijderd en daarmee wordt de plaats die het bestand op de schijf inneemt, vrijgegeven.

Dat zou niet erg zijn als DOS bij een volgende schrijfhandeling bij de eerste byte van een niet gebruikt deel zou beginnen. Om allerlei praktische redenen gebeurt dat niet: opeenvolgende bestanden sluiten nooit naadloos aan. DOS werkt met sectoren (delen van 512 bytes) en *clusters*. Een cluster is de kleinste eenheid die DOS kan lezen of schrijven. Hoe groot een cluster is, hangt af van het type schijf: op een schijf met een capaciteit van 360 Kb bevat een cluster twee sectoren (dus 1024 bytes), op een schijf met een capaciteit van 720 Kb is een cluster één sector groot.

Beide factoren bij elkaar zorgen ervoor dat een bestand dat uit slechts één teken, dus één byte, bestaat op dergelijke schijven respectievelijk 1023 en 511 bytes ongebruikt laat. Als die bytes eerder in gebruik waren bij een bestand dat de sectoren wel helemaal opvulde, dan blijft daar nu nog veel leesbaars van over: de zogenaamde *slack*. Met een programma als PC TOOLS kunt u dit allemaal rustig eens bekijken met de opdracht Hex Edit File van het menu File.

10 Tips en trucs voor het toetsenbord

MS-DOS kent talloze functies waarmee u het toetsenbord kunt aanpassen aan uw eigen wensen en behoeften. Van eenvoudige waarmee u de snelheid van de cursor aanpast tot wat meer gecompliceerde waarmee u de functie van toetsen verandert.

10.1 De snelheid van het toetsenbord wijzigen

Zelfs op een snelle AT of een 80386 kan de cursor soms hinderlijk langzaam zijn. Vanaf MS-DOS 4.0 kunt u die snelheid wijzigen via de opdracht MODE.

Uitwerking

Typ achter de systeemprompt de opdracht MODE met twee parameters; de eerste voor de snelheid, de tweede voor de wachttijd:

```
MODE CON RATE=snelheid DELAY=wachttijd
```

Geldige waarden voor de snelheid liggen tussen 1 (langzaam) en 32 (snel). De wachttijd bepaalt hoe lang u een toets ingedrukt moet houden voor hij wordt herhaald. Bij verstek is de wachttijd 0,25 seconde; geldige waarden liggen tussen één en vier. De hoogste snelheid die u uit uw toetsenbord kunt halen, bereikt u met de volgende opdracht:

```
MODE CON RATE=32 DELAY=1
```

Wellicht meldt het systeem u dat de computer deze waarden niet ondersteunt:

```
Function is not supported on this computer
```

In dat geval kunt u de snelheid niet wijzigen.

Wilt u de nieuwe instelling blijven gebruiken in de toekomst, dan neemt u de opdracht op in uw bestand AUTOEXEC.BAT.

Toelichting

Een AT-toetsenbord heeft een eigen kleine micro-processor, die in principe programmeerbaar is. Daardoor hoeft de computer zich niet meer om het toetsenbord te bekommeren; hij krijgt de gegevens kant en klaar van het toetsenbord aangeleverd. Dat geldt ook voor de herhalingssnelheid en de wachttijd.

Er kunnen twee redenen zijn die verhinderen dat het op uw computer werkt:

1 Het toetsenbord is niet 100 procent compatibel en begrijpt de opdrachten van de computer niet, zodat MS-DOS de beoogde waarden niet kan instellen.

2 U hebt geen AT, maar een XT. Het toetsenbord van een XT is niet programmeerbaar. U kunt de verstekwaarden niet wijzigen.

10.2 NumLock uitschakelen

Veel BIOS-versies (vooral bij AT's) schakelen NumLock automatisch in bij een reset, zodat het numerieke toetsenbord alleen voor de invoer van cijfers geschikt is. De cursorfuncties van de toetsen zijn uitgeschakeld. Met een klein programmaatje in machinetaal kunt ervoor zorgen dat u niet steeds op de toets NumLock hoeft te drukken om die situatie weer te veranderen.

Uitwerking

Het onderstaande programma NUMLOCK.COM schakelt op AT's en 80386-computers tevens de LED-indicator uit. Op XT's blijft die branden en geeft dan dus precies het tegenovergestelde aan. Voer de programmaregels met een editor in onder de naam NUMLOCK.DEB.

```
A
MOV AX,0040
MOV DS,AX
MOV AL,[0017]
XOR AL,20
MOV [0017],AL
RET

RCX
000E
```

```
nNUMLOCK.COM
W
Q
```

Vergeet de witregel na de opdracht RET niet, hij speelt een wezenlijke rol. Converteer de programmaregels met de debugger naar machinetaal:

```
DEBUG <NUMLOCK.DEB
```

Op uw scherm verschijnt de volgende listing:

```
-A
14A8:0100 MOV AX,0040
14A8:0103 MOV DS,AX
14A8:0105 MOV AL,[0017]
14A8:0108 XOR AL,20
14A8:010A MOV [0017],AL
14A8:010D RET
14A8:010E
-RCX
CX 000
:000E
-nNUMLOCK.COM
-W
Writing 0000E byte
-Q
```

In plaats van het adres '14A8' kan uw systeem een andere waarde uitvoeren. Met de opdracht DEBUG hebt u het kant en klare programma NUMLOCK.COM aangemaakt, dat dezelfde schakelfunctie heeft als de NumLock-toets. Kopieer het programma naar uw directory C:\DOS en neem de opdracht het te starten op in uw AUTOEXEC.BAT, zodat uw systeem automatisch de cursorbesturingstoetsen van het numerieke blok activeert bij een reset.

Toelichting

De status van de drie LEDs NumLock, CapsLock en ScrollLock wordt door het BIOS bewaard op geheugenadres 0040:0017. Elk van de drie toetsen heeft een bit toegewezen gekregen. Als die bit is gezet, is de functie ingeschakeld. De hieronder staande tabel geeft de toetsen en hun waarden weer.

toets	bit	hex	dec
NumLock	6	20	32
CapsLock	5	40	64
ScrollLock	4	10	16

Het programma in machinetaal haalt de waarde van het geheugenadres 0040:0017 op en draait de waarde van bit 5 om. Vervolgens wordt de gewijzigde waarde teruggeschreven naar het geheugenadres.

Tip: De functie van de afzonderlijke toetsen wordt niet bepaald door de toetsenbord-driver die u hebt geladen (KEYB). Hebt u in het bestand CONFIG.SYS de device driver ANSI.SYS geladen, dan kunt u de functie van afzonderlijke toetsen naar behoefte wijzigen. Dat is vooral zinvol voor toetsen en toetsencombinaties die u gewoonlijk niet gebruikt of die zelfs helemaal geen functie hebben.

10.3 Shortcuts

Directories activeren in DOS is lastig. U werkt in een diepvertakte directory (C:\PROGS\MIJNPROG\UTILS\TEKSTEN), kopieert even iets naar een andere net zo diepvertakte directory, wilt controleren of de bestanden daar zijn aangekomen en vervolgens verder werken in de oorspronkelijke directory. Met de opdracht CD blijft u typen. In paragraaf 4.14 hebben we u een oplossing aan de hand gedaan waar we nu nog een keer op terugkomen.

Uitwerking

In paragraaf 4.14 hebben we twee kleine batchbestanden gebruikt, CD_SET en CD_GET, waarmee het zoekpad naar de actuele directory werd opgeslagen in een systeemvariabele CURRENTDIR. Met behulp van die variabele kon u dan weer terugkeren naar de oorspronkelijke directory.

Het vergt bij die oplossing nog steeds wat typewerk om de opdrachten CD_SET en CD_GET in te voeren. MS-DOS gebruikt normaal gesproken de functietoetsen F11 en F12 niet en die zijn dus heel geschikt om dat werk over te nemen. In het bestand CONFIG.SYS moet dan wel de device driver ANSI.SYS zijn geladen met een opdracht als:

```
DEVICE = ANSI.SYS
```

Zie daarvoor ook hoofdstuk 7. Het onderstaande batchbestand F11_F12.BAT wijst de opdrachten toe aan de functietoetsen F11 en F12. De beide batchbestanden en de nodige hulpprogramma's moeten wel beschikbaar zijn.

```
@REM F11_F12.BAT
@ECHO ON
PROMPT $e[0;133;"CD_SET";13p
PROMPT $e[0;134;"CD_GET";13p
PROMPT $P$G
```

U kunt deze regels natuurlijk ook toevoegen aan het bestand AUTOEXEC.BAT of het batchbestand van daaruit oproepen met de opdracht CALL. De opdracht @ECHO ON hebben we bewust toegevoegd, omdat opdrachten die bedoeld zijn voor de driver ANSI.SYS niet werken als ECHO is uitgeschakeld.

Een constructie waarbij een opdracht wordt weggestopt achter een enkele toetsaanslag, is een *shortcut*. Met behulp van de device driver ANSI.SYS en de opdracht PROMPT kunt u hele reeksen van opdrachten voorzien van een shortcut. Welke dat zijn hangt natuurlijk erg af van de vraag hoe vaak u een opdracht gebruikt.

Bij het maken van shortcuts dient u wel rekening te houden met de oorspronkelijke functie van toetsen. Werkt u met DOSKEY, dan komen eigenlijk de functietoetsen F1 tot en met F6 vrij: u kunt nu bijvoorbeeld met cursor omhoog een opdracht uit de buffer terughalen (in plaats van met F3). Op een AT-toetsenbord beschikt u over twee series cursortoetsen; van die van het numerieke toetsenbord zou u shortcuts kunnen maken.

Daarnaast bieden de Alt-toets, de Shift-toets en de Ctrl-toets mogelijkheden te over om combinaties te maken. Ook in dat geval dient u wel eerst na te gaan of zo'n combinatie niet al in gebruik is. De SETUP-procedures van het Phoenix BIOS is bijvoorbeeld weggestopt achter de toetsencombinatie Ctrl-Alt-S.

Het hieronder staande batchbestand SHORTCUT.BAT bestaat uit een aantal vrij willekeurig gekozen functiewijzigingen van toetsen, bedoeld als voorbeeld. Het is een uitbreiding van het batchbestand hierboven. Omdat sommige applicaties niet met de shortcuts uit de voeten kunnen of de toetsencombinaties voor eigen doeleinden gebruiken, hebben we een tweede batchbestand gemaakt, waarin de oude toestand wordt hersteld: SHORTNOT.BAT.

SHORTCUT.BAT

```
@REM **************************************************
@REM * SHORTCUT.BAT - functies van toetsen aanpassen *
@REM **************************************************
SET OLDPROMPT=%PROMPT%
PROMPT $e[0;133;"CD_SET";13p
@REM directory opslaan in systeemvariabele: F11
PROMPT $e[0;134;"CD_GET";13p
@REM directory activeren vanuit systeemvariabele: F12
PROMPT $e[0;48;"BACKUP *.* A: /S /M"p
@REM reservekopieën maken van gewijzigde bestanden: Alt-B
PROMPT $e[0;32;"DIR /P";13p
@REM DIR per pagina: Alt-D
PROMPT $e[0;33;"FORMAT A:"p
@REM diskette in A: formatteren, bevestigen met Enter: Alt-F
PROMPT $e[0;50;" | MORE";13p
SET PROMPT=%OLDPROMPT%
```

SHORTNOT.BAT

```
@REM ****************************************************************
@REM * SHORTNOT.BAT - oorspronkelijke functies van toetsen          *
@REM * herstellen                                                   *
@REM ****************************************************************
SET OLDPROMPT=%PROMPT%
PROMPT $e[0;133;0;133p
PROMPT $e[0;134;0;134p
PROMPT $e[0;48;0;48p
PROMPT $e[0;32;0;32p
PROMPT $e[0;33;0;33p
PROMPT $e[0;50;0;50p
SET PROMPT=%OLDPROMPT%
```

Met behulp van beide batchbestanden kunt u nu de functie van een aantal toetsen aanpassen. De onderstaande tabel geeft daar een overzicht van.

toets	opdracht	toelichting
F11	CD_SET	actuele directory opslaan in systeemvariabele
F12	CD_GET	opgeslagen directory activeren
Alt-B	BACKUP A: /S /M	alle bestanden die zijn gewijzigd sinds de laatste backup opslaan, inclusief subdirectories
Alt-D	DIR /P	overzicht van de inhoud van de actuele directory per pagina, direct uitvoeren
Alt-33	FORMAT A:	diskette in drive A: formatteren, bevestigen met Enter
Alt-50	| MORE	uitvoer per pagina

Toelichting

Alle ANSI-escape-reeksen worden afgesloten met de letter 'p'. Opvallend is dat sommige opdrachten het getal '13' is toegevoegd voor de p. Daarmee wordt de toetsaanslag Enter nagebootst, zodat een opdracht direct wordt uitgevoerd. Bij onschuldige opdrachten als DIR is dat erg handig; bij gevaarlijke opdrachten als FORMAT A: kunt u hem beter achterwege laten.

Ten slotte geven we u hieronder een overzicht van de toetscodes die u kunt gebruiken in de escape-reeksen.

toets	code	toets	code	toets	code	toets	code
F1	59	Alt-F1	104	Shift-F1	84	Ctrl-F1	94
F2	60	Alt-F2	105	Shift-F2	85	Ctrl-F2	95
F3	61	Alt-F3	106	Shift-F3	86	Ctrl-F3	96
F4	62	Alt-F4	107	Shift-F4	87	Ctrl-F4	97
F5	63	Alt-F5	108	Shift-F5	88	Ctrl-F5	98
F6	64	Alt-F6	109	Shift-F6	89	Ctrl-F6	99
F7	65	Alt-F7	110	Shift-F7	90	Ctrl-F7	100

toets	code	toets	code	toets	code	toets	code
F8	66	Alt-F8	111	Shift-F8	91	Ctrl-F8	101
F9	67	Alt-F9	112	Shift-F9	92	Ctrl-F9	102
F10	68	Alt-F10	113	Shift-F10	93	Ctrl-F10	103
F11	133	Alt-F11	139	Shift-F11	135	Ctrl-F11	137
F12	134	Alt-F12	140	Shift-F12	136	Ctrl-F12	138
Alt-A	30	Alt-H	35	Alt-O	24	Alt-V	47
Alt-B	48	Alt-I	23	Alt-P	25	Alt-W	17
Alt-C	46	Alt-J	36	Alt-Q	16	Alt-X	45
Alt-D	32	Alt-K	37	Alt-R	19	Alt-Y	44
Alt-E	18	Alt-L	38	Alt-S	31	Alt-Z	21
Alt-F	33	Alt-M	50	Alt-T	20		
Alt-G	34	Alt-N	49	Alt-U	22		
Alt-1	120	Alt-4	123	Alt-7	126	Alt-0	129
Alt-2	121	Alt-5	124	Alt-8	127	Alt- -	130
Alt-3	122	Alt-6	125	Alt-9	128	Alt- =	131
links	75	Ctrl-links	115	Home	71	Ctrl-Home	119
rechts	77	Ctrl-rechts	116	End	79	Ctrl-End	117
omhoog	72			PgUp	73	Ctrl-PgUp	132
omlaag	80			PgDn	81	Ctrl-PgDn	118
Ins	82	Del	83	Shift-Tab	15	Ctrl-PrtScr	114

Tip: De functietoetsen F11 en F12 zijn pas vanaf MS-DOS 4.0 beschikbaar. Maak het batchbestand SHORTCUT.BAT niet al te groot; voor de device driver ANSI.SYS is een bepaalde geheugenruimte gereserveerd. Overschrijdt u die geheugenruimte, dan kunnen er rare dingen gebeuren.

11 Tips en trucs voor de MS-DOS Shell

MS-DOS 4.0 was de eerste versie van DOS met een DOS Shell. Het spreekt dus vanzelf dat daar nog wel wat aan te verbeteren viel. In versie 5.0 is dat dan ook gedaan. In dit hoofdstuk geven we u een aantal aanwijzingen voor het gebruik van de DOS Shell.

11.1 De Task swapper laden zonder wachten

Als u de *Task swapper* van de DOS Shell gebruikt, weet u dat het nogal wat tijd kost om de programma's op te nemen in de *Active task list*, de lijst met actieve programma's. U start een programma, wacht tot het is geladen en schakelt met Ctrl-Esc terug naar de DOS Shell. Vervolgens start u een tweede programma, wacht weer en keert weer terug, enzovoort. Met behulp van de Shift-toets kan het sneller.

Uitwerking

Als u een programma start met Shift-Enter in plaats van Enter, wordt het programma wel opgenomen in de lijst van actieve programma's, maar nog niet geladen en gestart. Pas als u het vanuit de lijst oproept, wordt het geladen en gestart. Vanzelfsprekend moet de Task swapper in het menu Options zijn geactiveerd: er staat dan een punt voor de optie. Werkt u met de muis, dan houdt u de Shift-toets ingedrukt, terwijl u klikt om hetzelfde resultaat te bereiken.

Het grote voordeel is dat u niet tijdenlang wacht tot alle programma's zijn opgenomen in de lijst met actieve programma's, terwijl achteraf blijkt dat u bijvoorbeeld maar twee van de zeven echt hebt gebruikt. Het kost even weinig moeite een programma weer te verwijderen uit de lijst: u markeert het programma in de lijst en drukt op de Del-toets. De DOS Shell stelt u nog een controlevraag en daarna wordt het programma uit de lijst verwijderd.

11.2 Startparameters toestaan

Als u in de Program Manager groepen en programma's opneemt, kunt u snel programma's starten: u hoeft slechts op een toets te drukken. De DOS Shell staat toe dat u voor de start van het programma nog parameters aan de opdracht toevoegt.

Uitwerking

In het onderstaande voorbeeld hebben we de tekstverwerker WORD aan de lijst toegevoegd, inclusief de mogelijkheid het programma te starten en tegelijk een tekstbestand te openen. U gaat als volgt te werk om een programma toe te voegen aan de lijst van de Program manager, zodanig dat u bij het starten ervan parameters op kunt geven. U kiest in het menu View de optie Program List. Vervolgens kiest u in het menu File de optie New. Er wordt een dialoogvenster geopend waarin u kunt kiezen of u een programma of een groep van programma's toe wilt voegen. U kiest Program Item. Evenals in batchbestanden kunt u in de Program Manager plaatshouders voor variabelen aangeven met een procentteken. In het nu geopende dialoogvenster voert u de volgende gegevens in:

```
Program Title:          tekstverwerker WORD
Commands:               WORD %1
Startup Directory:      C:\WORD
[ ] Pause after exit
```

In het voorbeeld gaan we ervan uit dat de tekstverwerker is opgeslagen in de directory C:\WORD. Wellicht hebt u uw harde schijf anders ingedeeld. In dat geval zult u ook een ander zoekpad moeten opgeven bij Startup Directory.

De plaatshouder '%1' geeft aan de DOS Shell door dat er aan de opdracht een parameter moet worden gekoppeld. Bij het sluiten van het dialoogvenster wordt er dan ook een nieuw dialoogvenster geopend, waarin u die parameters kunt opgeven:

```
Window Title:           WORD programmastart
Program Information:    Voer een bestandsnaam in
Prompt Message:         Bestandsnaam:
Default Parameters:     /l
```

De optie /l is een specifieke WORD-optie, die automatisch de tekst laadt die als laatste is bewerkt. De optie heeft op zich dus niets met de DOS Shell te maken.

De 'tekstverwerker WORD' maakt nu deel uit van de lijst van de Program Manager. Als u het start door het te markeren en op Enter te drukken, verschijnt er een extra dialoogvenster waarin u een bestandsnaam kunt opgeven. Drukt u op Enter zonder een bestandsnaam in te voeren, dan treedt de optie /L in werking en wordt het bestand geladen waaraan u het laatst hebt gewerkt. In het dialoogvenster kunt u die optie ook wissen met de Del-toets.

11.3 Programma's starten met een gegevensbestand

Vanuit de Dos Shell kunt u heel snel programma's starten met een gegevensbestand dat met het programma moet worden bewerkt. Daartoe richt u de Program Manager op een speciale manier in.

Uitwerking

Markeer in de eerste plaats het programmabestand dat u samen met een gegevensbestand wilt starten. Open dan het menu File en kies de optie Properties.... In plaats van de optie /L voert u bij de regel 'Default Parameters' de plaatshouder %F in. De letter F staat voor File (bestand).

Open het menu View en wijzig de weergave op het scherm; kies de optie Program/File List. Het scherm wordt in tweeën gedeeld: in het bovenste venster staan uw bestanden, in het onderste de programma's, waaronder ook 'tekstverwerker Word'. Markeer in het bovenste venster een gegevensbestand dat u wilt bewerken, bijvoorbeeld BRIEF.TXT en activeer vervolgens in het onderste venster de tekstverwerker. De Program Manager opent nu eerst een dialoogvenster waarin BRIEF.TXT al is ingevuld als bestand om te bewerken. U hoeft die opgave alleen nog maar te bevestigen met Enter. Eventueel kunt u in dat dialoogvenster een ander bestand opgeven of de optie /L invoeren.

Het handige van de optie %F is dat u nu een gegevensbestand kunt laden in een programma zonder dat u een volledig zoekpad hoeft te typen. Een klik met de muis volstaat om het gegevensbestand te markeren.

Tip: Tussen neus en lippen door hebben we hierboven al aangegeven dat u voor het functioneren van de procedure voor de weergave van programma's en bestanden dient te kiezen. Als u alleen programma's laat weergeven op uw scherm, wordt na het starten van een programma het dialoogvenster voor het invoeren van een gegevensbestand geopend zonder dat er iets is ingevuld.

11.4 Programma's laden met het laatst bewerkte bestand

Als u gecharmeerd bent van de mogelijkheid van WORD om met de optie /L het laatste door u bewerkte bestand opnieuw te laden, biedt de DOS Shell u een mogelijkheid ook andere programma's op die manier te starten. Daartoe dient de optie %L.

Uitwerking

Markeer in het venster MAIN van de Program Manager het programmabestand, bijvoorbeeld 'tekstverwerker WORD'. Open vervolgens het menu File en kies Properties. Bij Commands geeft u nu eerst de naam van het programmabestand op, gevolgd door de optie %1. Bevestig de invoer met Enter en vul in het volgende dialoogvenster bij Default Parameters %L in. Sluit ook dit venster met Enter.

De eerste keer dat u nu het programma start, gebeurt er niets bijzonders. De Program Manager kent nog geen laatst bewerkt bestand en opent dus een dialoogvenster waarin u een gegevensbestand invoert, eventueel voorafgegaan door een zoekpad, bijvoorbeeld:

```
C:\TEKSTEN\BRIEF.TXT
```

Het programma start nu met het gegevensbestand. Beëindigt u het programma en start u het daarna nogmaals op, dan opent het dialoogvenster voor het invoeren van een gegevensbestand met de naam van het bestand dat u zo even hebt bewerkt, al ingevuld. U hoeft die opgave nog slechts te bevestigen. Dit werkt ook als in de lijst met bestanden van de Program Manager een ander gegevensbestand is gemarkeerd.

Tip: De procedure blijft werken, ook nadat u de DOS Shell verlaat. De eerstvolgende keer dat u de DOS Shell activeert en het programma start, zal het gegevensbestand al worden ingevuld in het dialoogvenster. Dat komt omdat de DOS Shell de bestandsnaam opslaat in het bestand DOSSHELL.INI, waarin ook de instellingen worden bewaard van de DOS Shell.

Beide opties /F en /L laten zich uiteraard niet combineren. De Program Manager zou niet meer weten welke bestandsnaam als verstekwaarde moest worden ingevuld in het dialoogvenster. Voor WORD is dit probleem op te lossen. Als u een bestand hebt gemarkeerd, start u WORD met dat bestand; zo niet, dan start u WORD met het laatst bewerkte bestand. Om dat te bewerkstelligen opent u opnieuw het menu File en kiest Properties.... In het dialoogvenster voert u bij Commands de volgende opdracht in:

```
IF NOT "%1"=="" WORD %1 ; IF "%1"=="" WORD/L
```

Bevestig de invoer met Enter. In het tweede dialoogvenster vult u bij Default Parameters de optie %F in.

Start u nu WORD, dan is in het dialoogvenster voor het invoeren van het te openen

gegevensbestand het gemarkeerde bestand ingevuld. Drukt u op Enter, dan wordt dat bestand geladen. Druk u op Del, dan wordt die bestandsnaam in het dialoogvenster gewist, waarna u met Enter automatisch het laatst bewerkte bestand oproept.

Tip: Wis de bestandsnaam niet met de Backspace-toets of de spatiebalk, dan werkt de truc niet. Drukt u daarna namelijk op Enter, dan verschijnt opnieuw het dialoogvenster waarin u een naam moet invullen. Het is zelfs mogelijk dat er iets geheel anders gebeurt.

11.5 De DOS Shell en de muis

Met behulp van de muis kunt u een aantal handelingen in de DOS Shell snel uitvoeren zonder dat u nog van alles hoeft in te voeren via het toetsenbord. Met name bij het kopiëren of verplaatsen van bestanden biedt dit grote voordelen.

Uitwerking

Zo lang u aan het experimenteren bent met de mogelijkheden die we in deze paragraaf bespreken, raden we u aan de functie Confirm on Mouse Operation in te schakelen. Open het menu Options en kies Confirmation. Staat er een 'X' voor de functie, dan is hij actief. Met de muis werkt u namelijk zo snel, dat u voordat u het door hebt opdrachten hebt gegeven, die heel iets anders doen dan u van plan was. U staat raar te kijken als het systeem plotseling start zonder de gebruikelijke configuratie. Na lang peinzen en zoeken komt u er dan achter dat het bestand AUTOEXEC.BAT is verdwenen uit de hoofd-directory en dat alles omdat u gisteren het bestand niet hebt gekopieerd naar de directory C:\TEST, maar verplaatst...

Om vergelijkbare redenen is het verstandig in het begin de functie Select Across Directories in het menu Options uit te schakelen (er staat dan geen punt voor).

Ten slotte raden we u aan om de muisfuncties uit te proberen, terwijl de Program Manager op uw scherm een Single File List weergeeft (in het menu View).

Een selectie van bestanden markeren

Er zijn twee methoden om met behulp van de muis snel een selectie van bestanden te markeren. Betreft het een aaneengesloten groep, dan klikt u met de muis op het eerste bestand, drukt de Shift-toets in en klikt op het laatste bestand. Alle tussenliggende bestanden worden nu geaccentueerd weergegeven en zijn dus geselecteerd. Wilt u her en der in een directory een aantal bestanden selecteren, dan houdt u de

Ctrl-toets ingedrukt, terwijl u met de muis op de gewenste bestandsnamen klikt.

Bestanden kopiëren

Wilt u een bestand kopiëren, dan plaatst u de muiscursor op de bestandsnaam. U drukt de linker knop van de muis in en houdt die ingedrukt. Vervolgens drukt u de Ctrl-toets in en houdt ook die ingedrukt. Als u nu de muis versleept, verandert de muiscursor, afhankelijk van de positie op het scherm. Staat de muiscursor op een 'toegestaan doel', dat wil zeggen een directory-naam, een andere bestandsnaam of een drive-letter, dan wordt de muiscursor weergegeven als een punt; staat hij op een andere positie, dan verandert hij in een dubbel uitroepteken.

Is de DOS Shell actief in een grafische modus, dan verandert de muiscursor respectievelijk in een pictogram voor 'bestand' (een archiefkaartje met omgevouwen hoek) of een verbodsbord ('Verboden te parkeren'). Zodra u het doel hebt bereikt, laat u de muisknop en de Ctrl-toets los.

Bestanden verplaatsen

U verplaatst bestanden op dezelfde manier als u bestanden kopieert, met dien verstande dat u nu de Ctrl-toets niet indrukt. Houdt u bij beide handelingen de statusregel onder in het scherm in de gaten; die geeft aan welke functie actief is.

Tip: Als de doel-directory bij het kopiëren of verplaatsen van bestanden niet wordt weergegeven in het venster met de Directory Tree, kunt u die in beeld brengen door de muiscursor - nog altijd met ingedrukte muisknop - op een van de beide pijlen van de schuifbalk te plaatsen. het venster met directories scrollt dan in de aangegeven richting. Het is belangrijk dat u de muisknop ingedrukt houdt, zo lang het doel niet is geaccentueerd.

Programma's starten samen met een gegevensbestand

In de vorige paragraaf hebben we een mogelijkheid besproken om een programma te starten samen met een gegevensbestand. Met de muis kan dat nog veel makkelijker. U plaatst de muiscursor op een gegevensbestand, drukt op de linker knop en houdt die ingedrukt, terwijl u de muiscursor versleept naar het programmabestand dat u wilt starten. Kijk naar de statusregel onder in beeld, daar ziet u terwijl u de muiscursor versleept, vermeld welk programmabestand zal worden geopend samen met het gegevensbestand. Als u het juiste programmabestand hebt bereikt, laat u de muisknop weer los.

Tip: Deze functie werkt het best als u in het menu File de optie Dual File List activeert. Het scherm wordt dan verdeeld over twee boven elkaar geplaatste vensters. In beide vensters worden een *Directory Tree* en een bestandenlijst weergegeven. In het ene venster activeert u de directory met het gegevensbestand, in het andere de die met het programmabestand.

11.6 De functie Associate

De DOS Shell maakt het mogelijk een of meer bestanden te kopiëren met een enkele dubbelklik van de muis. Of u kunt een tijdelijk bestand wissen met een enkele klik. Daarvoor hebt u de functie Associate uit het menu File nodig.

Uitwerking

Normaal gesproken kunt u in MS-DOS alleen programma's starten met een extensie .COM, .EXE of .BAT. In de Program Manager van de DOS Shell kunt u echter vastleggen welk programmabestand moet worden gestart als u een gegevensbestand met een bepaalde extensie kiest. De extensie wordt dan gekoppeld (geassocieerd) met een programmabestand. Het lijkt dan plotseling alsof u ook bestanden met de extensie .TXT zelfstandig kunt starten.

Tip: U kunt bestanden met een extensie .COM, .EXE of .BAT niet zelf koppelen aan een ander programmabestand. Zou u dat toch proberen, dan start de Program Manager het eerste bestand, dat u dus eigenlijk als gegevensbestand beschouwt. Het tweede programmabestand wordt genegeerd.

In de DOS Shell zijn al koppelingen gemaakt: dubbelklikt u op een bestand met de extensie .TXT, dan wordt automatisch het programma EDIT geladen met het gekozen gegevensbestand.

U koppelt een gegevensbestand aan een programmabestand door het programmabestand te markeren en vervolgens in het menu File de optie Associate... te kiezen. In het dialoogvenster vult u dan de extensie in van de gegevensbestanden waarmee u het programmabestand wilt kunnen openen.

Veel trucs onder MS-DOS zijn het resultaat van doordenken over de voorgeprogrammeerde mogelijkheden. Dat is ook hier het geval. Een kleine aanpassing van de koppeling maakt het mogelijk een bestand te wissen of te kopiëren via een dubbelklik.

Bestanden kopiëren met een dubbelklik

We gaan ervan uit dat u liever niet automatisch de editor EDIT start, als u dubbelklikt op een gegevensbestand met de extensie .TXT. In plaats daarvan wilt dergelijke bestanden kopiëren. Daarvoor hebt u het volgende kleine batchbestand nodig, dat u onder de naam ASSCOPY.bat (ASSOCIATE_COPY.BAT) opslaat in de directory C:\BAT.

```
@REM ASSCOPY.BAT - kopieert een gemarkeerd bestand naar A:
@COPY %1 A:
@ECHO %1 is naar A: gekopieerd
```

Vervolgens markeert u in de DOS Shell een bestand met de extensie .TXT en kiest in het menu File de optie Associate. In het dialoogvenster geeft de Program Manager de bestaande koppeling aan. U vervangt die door 'C:\BAT\ASSCOPY.BAT'. Plaats daarna een diskette in drive A: en dubbelklik op een willekeurig bestand met de extensie .TXT. Het bestand wordt gekopieerd naar A:.

Een tijdelijk bestand met een klik wissen

Sommige applicaties maken tijdelijke bestanden met de extensie .TMP. Beëindigt u het programma op de gangbare manier, dan worden ze gewist, maar is de computer blijven hangen, dan staan ze nog steeds op schijf. Als u zo'n bestand snel wilt wissen, markeert u het en geeft u in het dialoogvenster van Associate de opdracht DEL.

Een dubbelklik op de bestandsnaam is daarna voldoende om het te wissen. In de bestandenlijst van de DOS Shell blijft het nog aanwezig tot u de Program Manager opdracht geeft die lijst up to date te maken met een dubbelklik op de drive-letter.

11.7 Bestanden verdelen over meerdere diskettes

De hoeveelheid gegevens en de omvang van bestanden neemt steeds meer toe. Er staan vaak zo veel gegevens in één directory, dat de opdracht die naar een schijf te kopiëren strandt met een foutmelding als de schijf vol is. Naast BACKUP kent de DOS Shell nog een mogelijkheid voor het verdelen van bestanden over schijven.

Uitwerking

U markeert eerst de bestanden en begint vervolgens met kopiëren. Als de DOS

Shell u met een foutmelding meldt dat de diskette vol is, kiest u in het dialoogvenster voor '1. Skip this file or directory and continue'. U noteert de naam van het bestand dat wordt overgeslagen. Ondertussen hebt u een andere schijf in de drive gedaan en gaat u verder met kopiëren. Op de laatste diskette dient u ten slotte zelf nog de overgeslagen bestanden te kopiëren.

11.8 Lange opdrachten definiëren in de DOS Shell

Als u een nieuw programma toevoegt aan de bestandenlijst van het venster Main van de Program Manager, krijgt u de gelegenheid DOS-opdrachten in te voeren. De afzonderlijke opdrachten scheidt u met puntkomma's. U mag maximaal 255 tekens gebruiken. Dat lijkt erg veel, maar als tussen die opdrachten ook een of meer ECHO-opdrachten zitten, is 255 tekens eigenlijk erg weinig. U lost het probleem op door batchbestanden te gebruiken.

Uitwerking

Omdat batchbestanden vrijwel onbeperkt lang mogen zijn, kunt u de opdrachten bij het programma er altijd in kwijt.

U kunt het batchbestand in de DOS Shell-opdrachtregel niet rechtstreeks oproepen met de naam van het batchbestand. Als u dat doet, stopt de uitvoering na het einde van het batchbestand; overige opdrachten worden daarna genegeerd. U dient daarvoor de opdracht CALL te gebruiken. Voor de oproep van het batchbestand FINDALL.BAT met de parameter .TXT luidt de opdracht:

```
CALL FINDALL .TXT
```

De opdrachtregels van de DOS Shell werken dus net als een batchbestand en zijn ook aan dezelfde regels gebonden. Dat betekent dat u ook andere opdrachten die u in batchbestanden gebruikt, bijvoorbeeld IF, in de opdrachtregel van de DOS Shell kunt gebruiken. Een voorbeeld daarvan ziet u in paragraaf 11.4.

11.9 Problemen met de Task swapper

Gebruikt u de Task swapper, dan kunt u geconfronteerd worden met een aantal bijzondere problemen. Voordat we daar wat over kunnen zeggen, zullen we wat dieper in moeten gaan op de werking van de Task swapper.

Uitwerking

U activeert de Task swapper door de optie **Enable Task Swapper** in het menu Options te kiezen. De onderste helft van uw scherm wordt in twee helften verdeeld: links houdt u een overzicht van groepen en programma's van de Program Manager; rechts verschijnt een nieuw venster met de titel **Active List**. Aanvankelijk is dat venster leeg, maar iedere keer dat u een programma start en vervolgens verlaat met Ctrl-Esc, wordt het toegevoegd aan die lijst. De toetsencombinatie Ctrl-Esc zorgt ervoor dat het programma in zijn actuele toestand wordt 'ingevroren' op de harde schijf. Een dubbelklik op de bestandsnaam in de lijst met actieve programma's ontdooit het programma weer.

De DOS Shell slaat twee belangrijke gegevens op op de harde schijf:

- de door het programma bezette geheugenruimte
- informatie voor het starten van het programma.

Waar de DOS Shell die gegevens opslaat, hangt af van de vraag of u in het bestand AUTOEXEC.BAT een directory voor tijdelijke bestanden hebt aangelegd:

```
SET TEMP=C:\TEMP
```

Als u een paar programma's in de DOS Shell hebt gestart, kunt u die tijdelijke bestanden bekijken. Daartoe start u vanuit de DOS Shell een tweede besturingssysteem met een dubbelklik op **Command Prompt** in het venster **Main**. Met de opdracht **DIR C:\TEMP** roept u een overzicht als het volgende op het scherm:

```
 Volume in drive C has no label
 Volume Serial Number is 1738-069C
 Directory of C:\TEMP

.            <DIR>       30-07-91   23:07
..           <DIR>       30-07-91   23:07
DOSSHELL SWP      43841  27-10-91   13:55
ED03DOSC BAT         18  27-10-91   13:55
EF19DOSC BAT         19  27-10-91   13:55
        5 file(s)       43878 bytes
                       888832 bytes free
```

De DOS Shell maakt voor ieder programma dat u start vanuit de Program Manager een batchbestand aan. Dit kunt u met de opdracht **TYPE** bekijken. De inhoud van het bestand EF19DOSC.BAT hierboven is:

```
@C:\DOS\COMMAND
```

Het starten van het tweede besturingssysteem heeft dus geleid tot een batchbestand met de opdracht COMMAND. De apestaart zorgt ervoor dat de opdracht bij uitvoering niet op het scherm verschijnt.

Het opgeslagen programma kunt u ook bekijken, zij het dat dat in een bestand staat waarvan het attribuut Hidden is gezet. Geeft u de opdracht DIR /A:H, dan wordt het weergegeven:

```
 Volume in drive C has no label
 Volume Serial Number is 1738-069C
 Directory of C:\TEMP

~DOS100E TMP    251626 27-10-91   13:55
        1 file(s)     251626 bytes
                      886784 bytes free
```

Wat u met de informatie kunt doen

U kunt deze gegevens op verschillende manieren gebruiken. In de eerste plaats kunt u de batchbestanden bekijken die de DOS Shell heeft aangemaakt. Als u zelf een programma hebt toegevoegd aan de lijst van de Program Manager, dat niet functioneert zoals u dat hebt bedoeld, kan de inhoud van het batchbestand u misschien verder helpen. U kunt in het batchbestand de opdracht PAUSE toevoegen om terug te kunnen keren naar de DOS Shell. Activeert u in de DOS Shell in het menu Options de functie File Display Options en vervolgens de functie Display hidden/system files, dan kunt u zonder meer vanuit de Program Manager de verborgen bestanden bekijken met de opgeslagen programma's.

U mag in geen geval de bestanden in de directory C:\TEMP wissen: u bent dan de actuele gegevens van uw opgeslagen programma kwijt. Overigens hoeft u niet bang te zijn dat uw harde schijf volraakt met dergelijke bestanden als uw computer blijft hangen; de eerstvolgende keer dat u de DOS Shell start, worden de oude tijdelijke bestanden gewist.

Bestanden beveiligen voor tegenstrijdige opdrachten

Multi-tasking levert altijd het probleem op dat twee of meer programma's hetzelfde gegevensbestand willen bewerken. In het ergste geval geeft u in het ene programma

de opdracht een bestand te wissen, terwijl u het bestand in het tweede programma juist bewerkt.

In een netwerk zorgt de netwerk-driver ervoor dat dergelijke conflicten worden vermeden. De Task swapper van de DOS Shell is niet op een dergelijke manier beveiligd en dus kunnen bijvoorbeeld de volgende conflicten ontstaan:

- U wist per ongeluk een bestand in de directory C:\TEMP: uw computer blijft hangen of de DOS Shell functioneert niet meer.
- U werkt in WORD en hebt de functie 'automatisch opslaan' geactiveerd. Per ongeluk start u WORD vanuit de DOS Shell. WORD start met een verwijzing naar reservekopieën (namelijk die van de WORD-kopie die u zojuist hebt opgeslagen in C:\TEMP) en vraagt of de reservekopieën moeten worden geactiveerd. Achteloos drukt u op 'J'. Pas achteraf merkt u dat u daardoor de tekst in de oorspronkelijk geladen kopie van WORD hebt verminkt.

In beide gevallen kunt het conflict voorkomen met de DOS-opdracht SHARE. Neem de opdracht op in het batchbestand waarmee u de DOS Shell start. SHARE zorgt er vervolgens voor dat een tweede programma een gegevensbestand niet kan wijzigen.

11.10 Een backup maken van DOSSHELL.INI

Het bestand DOSSHELL.INI bewaart alle instellingen van de DOS Shell. Het is erg vervelend als u veel werk en tijd hebt gestoken in het configureren van de DOS Shell en het bestand om de een of andere reden verloren of beschadigd raakt. U dient dus regelmatig een backup van het bestand te maken.

Uitwerking

Neem in het batchbestand waarmee u de DOS Shell start een regel op om een kopie van het bestand DOSSHELL.INI te maken, bijvoorbeeld:

```
COPY DOSSHELL.INI DOSSHELL.OLD
```

Uiteraard is het nog beter als u de reservekopie opslaat op een diskette. Als dan het bestand DOSSHELL.INI op de harde schijf samen met andere bestanden verminkt raakt, kunt u met de volgende opdracht de schade herstellen:

```
COPY A:\DOSSHELL.OLD C:\DOS\DOSSHELL.INI
```

Een beschadiging van DOSSHELL.INI is minder onwaarschijnlijk dan u wellicht denkt. De DOS Shell wordt voortdurend aan nieuwe omstandigheden aangepast. De kans dat daarbij de computer een keer blijft hangen en het bestand DOSSHELL.INI wordt beschadigd, is redelijk groot.

Tip: U mag het bestand DOSSHELL.INI in geen geval vanuit de DOS Shell zelf wijzigen of vanuit een programma dat u hebt gestart in de DOS Shell. Het is weliswaar zeer aanlokkelijk om vanuit de DOS Shell even de editor EDIT te starten en een instelling in het bestand aan te passen, maar dat zal zeker tot een systeemfout en een beschadiging van het bestand leiden.

11.11 De DOS Shell aanpassen per gebruiker

De DOS Shell kan slechts één set instellingen bewaren. Dat betekent dat, als u niet de enige bent die van de DOS Shell op uw computer gebruik maakt, u steeds wordt geconfronteerd met een gewijzigd bestand DOSSHELL.INI. Evenals de andere gebruikers, trouwens. Dit probleem lost u op door meerdere kopieën van de DOS Shell te installeren.

Uitwerking

De oplossing is eigenlijk heel eenvoudig: u start steeds de DOS Shell met een uniek gebruikersnummer; de eerste gebruiker start de DOS Shell dus met de opdracht:

```
SHELL 1
```

Er wordt dan bij de eerste oproep van de DOS Shell een bestand DOSSHELL.INI aangemaakt onder de naam SHELL1.INI. Vervolgens wordt de DOS Shell met dit configuratiebestand geopend. Bij beëindiging van het werk met de DOS Shell wordt dit bestand opgeslagen. Een tweede gebruiker start de DOS Shell met een volgend gebruikersnummer:

```
SHELL 2
```

Bij deze werkwijze hoort een speciaal batchbestand dat controleert of er een gebruikersnummer is opgegeven en dat vervolgens het juiste INI-bestand activeert:

```
@REM SHELL.BAT - de DOS Shell oproepen met een persoonlijk
@REM            .INI-bestand
```

```
@ECHO OFF
IF NOT "%1"=="" GOTO OK
ECHO.
CLS
ECHO.
ECHO Ongeldige opdracht.
ECHO ******************
ECHO U dient uw gebruikersnummer in te voeren achter de opdracht
ECHO Syntaxis: SHELL gebruikersnummer
GOTO Einde
:OK
C:\
CD DOS
IF EXIST SHELL%1.INI GOTO OK1
COPY DOSSHELL.INI SHELL%1.INI >NUL
ECHO De nieuwe gebruiker %1 is geaccepteerd, de DOS Shell is
ECHO geconfigureerd met de standaardinstellingen.
:OK1
COPY SHELL%1.INI DOSSHELL.INI >NUL
ECHO De DOS Shell wordt gestart voor gebruiker %1
PAUSE
DOSSHELL.COM
ECHO De instellingen voor gebruiker %1 zijn opgeslagen
PAUSE
C:\
CD DOS
COPY DOSSHELL.INI SHELL%1.INI >NUL
:Einde
@ECHO ON
```

Tip: U kunt dit batchbestand gebruiken voor alle programma's die de standaardinstellingen opslaan in een .INI-bestand door overal waar sprake is van de DOS Shell de naam van dat programma in te voeren.

Tip: Maakt uw computer deel uit van een netwerk, dan werkt de oplossing niet, omdat wellicht op meerdere netwerkstations tegelijkertijd de DOS Shell actief is. Al die stations willen dan hun instellingen opslaan in het uiteindelijk enige (!) bestand DOSSHELL.INI.

12 Backups maken en bestanden herstellen

In dit hoofdstuk willen we u een aantal aanwijzingen geven om op de meest zekere maar toch gebruikersvriendelijke manier uw gegevens te beveiligen. Daarnaast gaan we in op de mogelijkheden bestanden te herstellen, als er onverhoopt toch iets mis mocht gaan. Eerst willen we echter een hardnekkig misverstand uit de weg ruimen, namelijk dat gegevens alleen verloren gaan als u per ongeluk de opdracht DEL *.* verkeerd gebruikt of uw harde schijf formatteert.

De kans dat u op die manier gegevens kwijtraakt is zelfs zeer gering, vergeleken met alle andere gevaren die u bedreigen. De opdracht DEL *.* leidt hooguit tot het verlies van bestanden in de actuele directory en de opdracht FORMAT C: kunt u altijd in een noodgeval nog afbreken zonder verlies van gegevens.

De werkelijke gevaren schuilen in een klein hoekje, zoals de onderstaande voorbeelden duidelijk maken.

U hebt problemen met uw harde schijf. Het lijkt erop dat de controller van de harde schijf niet goed functioneert, want u kunt nog wel bestanden lezen, maar geen gegevens meer naar de harde schijf schrijven. Omdat u uw PC geen moment kunt missen, brengt u zelf na een telefonische afspraak, de PC onmiddellijk naar de dealer. In de haast vergeet u de koppen van de harde schijf te parkeren. Bij de dealer aangekomen, weigert uw PC om ook nog maar iets met de harde schijf te doen: onderweg zijn de koppen tegen de harde schijf aangekomen, een zogenaamde *head crash*: uw harde schijf is onherstelbaar beschadigd en alle gegevens erop zijn definitief verloren gegaan.

Uw tekstverwerker blijft bij het opslaan van een tekstbestand plotseling hangen. Aanvankelijk lijkt dat niet ernstig, want u beschikt over een reservekopie van het bestand. Na een reset gedraagt MS-DOS zich merkwaardig en geeft een aantal foutmeldingen weer op het scherm. CHKDSK meldt maar liefst zo'n 2000 gebieden op de harde schijf die onleesbaar zijn en inderdaad zijn de meeste bestanden op de harde schijf onleesbaar. Vanzelfsprekend hebt u de laatste tijd verzuimd een backup te maken...

U hebt per ongeluk een bestand gewist op de harde schijf en weet ondertussen dat u zoiets weer kunt herstellen. Het handboek voor MS-DOS is niet bij de hand, maar in het overzicht dat u met HELP op uw scherm tovert, valt u de opdracht RECO-

VER op. U typt de opdracht, bevestigt de controlevraag en wacht af. In plaats van dat uw bestand is hersteld, meldt DOS u vervolgens

```
329 file(s) recovered
```

U bent van de regen in de drup geraakt. De opdracht DIR laat u zien waarom: alle bestanden op uw harde schijf zijn in de hoofddirectory terechtgekomen en wat erger is, ze zijn allemaal hernoemd: van FILE0001.REC tot FILE0329.REC. Van subdirectories is geen sprake meer.

De voorbeelden zijn niet verzonnen. Catastrofes van deze omvang komen dagelijks voor: uw bestanden zijn op de harde schijf even veilig als uw portemonnee in het karretje van de supermarkt.

Het maken van een volledige backup van een harde schijf kost tijd, veel tijd. Dat is een belangrijke reden waarom veel gebruikers het steeds maar weer uitstellen. Tot het te laat is natuurlijk. Daarom loont het de moeite de volgende paragrafen door te lezen en de tips te gebruiken die u tijd besparen, maar toch het gevoel geven dat er niet echt iets fout kan gaan.

12.1 Snel backups maken

In lang niet alle gevallen is het handig reservekopieën te maken met behulp van de opdracht BACKUP. De opdrachten COPY en XCOPY doen tot op zekere hoogte hetzelfde werk.

Een reservekopie van het laatst bewerkte bestand

De eenvoudigste methode om een reservekopie van een bestand of zelfs van een paar bestanden, te maken, is met de opdracht COPY. U hebt bijvoorbeeld een tekstbestand bewerkt en daarna opgeslagen. U verlaat de tekstverwerker en kopieert daarna het opgeslagen bestand eenvoudigweg naar een diskette. Afhankelijk van de grootte van het bestand kunt u volstaan met één diskette of zult u de tip van paragraaf 11.7 moeten gebruiken.

Deze werkwijze biedt een uitstekende beveiliging tegen ongelukken, maar kost nauwelijks tijd. Hij vergt wel discipline, maar u krijgt er zekerheid voor terug.

Reservekopieën van meerdere bestanden

De opdracht COPY heeft zijn beperkingen. Als u een paar uur achter elkaar hebt gewerkt en daarbij diverse bestanden hebt gewijzigd, zult u uit uw hoofd moeten weten van welke bestanden u met COPY een reservekopie dient te maken. U zou een batchbestand aan kunnen maken dat het werk van u over neemt, maar dat kan alleen als het altijd om dezelfde bestanden gaat.

Zo lang de bestanden nog zonder problemen op één diskette passen, kunt beter de opdracht XCOPY gebruiken. Als u namelijk een bestand wijzigt, wijzigt MS-DOS het archiveringsattribuut van dat bestand. Daar kunt u achteraf gebruik van maken. Wilt u bijvoorbeeld alle tekstbestanden in de directory C:\TEKSTEN die u hebt gewijzigd, op een diskette opslaan, dan geeft u de opdracht:

```
XCOPY C:\TEKSTEN\*.TXT A: /M
```

De optie /M geeft aan dat alleen de bestanden waarvan het archiveringsattribuut is gezet, moeten worden gekopieerd. Hij zorgt er tevens voor dat het archiveringsattribuut vervolgens weer op nul wordt gezet. Als de directory bovendien nog subdirectories bevat waarin zich ook bestanden bevinden die eventueel zijn gewijzigd, dan voegt u nog de optie /S toe.

De opdracht XCOPY biedt u op deze manier dezelfde voordelen als de opdracht COPY: snel backups maken en toch zeker weten dat u niet alleen van de harde schijf afhankelijk bent.

12.2 De juiste BACKUP op het juiste moment

De opdracht BACKUP van MS-DOS is niet de meest geschikte manier om een reservekopie van uw harde schijf te maken. Toch kunnen we u wel wat tips geven, waarmee u de opdracht wat beter kunt gebruiken.

Een backup van relevante gegevens

Hebt u een harde schijf met een capaciteit van 80 Mb, dan kan het maken van een backup al gauw meer dan een uur duren. Dat is zonde van de tijd, als van de helft van die 80 Mb al een backup is gemaakt. We raden u aan het volgende systeem te gebruiken:

1 Zorg om te beginnen voor een goede directory-structuur van uw harde schijf: maak twee directories waaronder u alle programma's opslaat: C:\PROGS en C:\DATA. Sla vooral gegevensbestanden consequent op in de directory C:\DATA; hieronder bij punt twee zult u zien waarom.

2 Maak een backup van de hele harde schijf met de opdracht:

```
BACKUP C:\*.* A: /S
```

Van alle bestanden, ook die in subdirectories, wordt nu een backup gemaakt. Zorg dat u voldoende diskettes bij de hand hebt. Ze hoeven nog niet geformatteerd te zijn; als het nodig is, zorgt BACKUP daarvoor.

Raakt om de een of andere reden de inhoud van uw harde schijf verloren, dan kunt u met de opdracht RESTORE de bestanden terugschrijven naar de harde schijf. In de volgende paragraaf komen we daar op terug.

3 In het vervolg hoeft u nog slechts een backup te maken van de directories met gegevensbestanden. U hoeft bijvoorbeeld niet bij iedere backup van uw directory C:\DOS een reservekopie te maken, omdat daar in principe niets in is veranderd.

Nu blijkt ook waarom de scheiding tussen C:\PROGS EN C:\DATA zo belangrijk is. U maakt eenvoudigweg een aanvullende backup van de directory C:\DATA en alle subdirectories van C:\DATA:

```
BACKUP C:\DATA\*.* A: /S
```

4 Als u nu alleen een backup maakt van die bestanden waarvan het archiveringsattribuut is gezet, bespaart u zich nog meer tijd en moeite. Dat kan met de volgende opdracht:

```
BACKUP C:\DATA\*.* a: /S /M /A
```

BACKUP vraagt u de laatste diskette van de oorspronkelijke backup in drive A: te doen en controleert vervolgens van welke bestanden een reservekopie moet worden gemaakt. Alle bestanden waarvan nog geen actuele reservekopie bestaat, worden gekopieerd naar de laatste diskette en eventueel aanvullende (lege) diskettes.

Deze methode werkt snel en veilig. Het enige nadeel is dat niet over oude versies van bestanden wordt heen geschreven. Nieuwe versies worden óók op een backup-diskette gezet, met als gevolg dat de stapel backup-diskettes steeds groter

wordt. Bij tijd en wijle dient u dus van de gegevensbestanden een schone, nieuwe backup te maken met de opdracht:

```
BACKUP C:\DATA\*.* A: /S
```

12.3 De juiste RESTORE op het juiste moment

Ook bij het terughalen van verloren gegane bestanden kunt u zich nogal wat moeite te besparen door op het juiste moment de opdracht RESTORE te gebruiken met bepaalde parameters.

Een enkel bestand herstellen

Als slechts een enkel bestand op de harde schijf verloren gaat, kunt u RESTORE oproepen om dat ene bestand van de reservekopieën te lezen. Laten we zeggen dat het om het bestand C:\DATA\TEKSTEN\BRIEF12.TXT gaat, dan voert u de volgende opdracht in:

```
RESTORE A: C:\DATA\TEKSTEN\BRIEF12.TXT
```

RESTORE vraagt u dan achtereenvolgens de backup-diskettes in drive A: te plaatsen, totdat het gezochte bestand is gevonden. Vervolgens wordt het gelezen en weer naar de harde schijf geschreven.

Meerdere bestanden herstellen

U mag bij de opdracht RESTORE ook wildcards gebruiken en dat maakt het mogelijk om als bijvoorbeeld alle bestanden van de directory C:\DATA\TEKSTEN\NOTA_S verloren zijn gegaan, met de opdracht

```
RESTORE A: C:\DATA\TEKSTEN\NOTA_S\*.*
```

alle bestanden van die directory terug te halen van de backup.

Let op! Geeft u de opdracht

```
RESTORE A: C:\DATA\*.*
```

dan worden alle gegevensbestanden van de backup teruggelezen. Een aantal daarvan is ongetwijfeld al achterhaald en u bent dan dus in één klap ook uw actuele bestanden kwijt.

We bespreken hieronder nog een paar bijzondere gevallen:

1 Als een groot gedeelte van de bestanden van de harde schijf verloren is gegaan, gebruikt u de optie /N om alleen bestanden terug te halen die niet meer op de harde schijf voorkomen:

```
RESTORE A: C:\DATA\*.* /N
```

2 Bestanden zijn niet definitief gewist, maar wel voorlopig: de eerste letter van de bestandsnaam is verwijderd, en u weet niet meer welke letter dat was. Gebruik de optie /P als volgt:

```
RESTORE A: C:\DATA\*.* /P
```

RESTORE vraagt u bij elk bestand waarvan het archiveringsattribuut is gezet, of het vervangen moet worden door de reservekopie.

3 In situaties waarin u niet precies weet wat u moet doen, kunt u de optie /D gebruiken. RESTORE zet dan een lijst met bestandsnamen op uw scherm die overeenkomt met de opgave in de opdracht, maar voert nog geen wijzigingen door op de harde schijf.

Tip: Als er bestanden verloren zijn gegaan, dient u uiteraard voorzichtig te zijn bij het herstellen van de schade. Voordat u door hebt wat er gebeurt, bent u nog een aantal bestanden in een directory kwijt. Maak daarom voordat u RESTORE oproept een reservekopie van de directory waar het om gaat. Gebeurt er dan iets bij het herstellen, dan beschikt u tenminste nog over een minst-verminkte-versie.

Als RESTORE niet werkt...

Het kan gebeuren dat u een bestand bent kwijt geraakt, RESTORE opdracht geeft de situatie te herstellen, op verzoek alle diskettes van de backup in de drive plaatst, en dat RESTORE uiteindelijk toch meldt dat er geen reservekopie is gevonden die in aanmerking komt.

Het meest voor de hand liggend is, dat u een onjuist zoekpad hebt gebruikt of de be-

standsnaam hebt verhaspeld of onjuiste wildcards hebt gebruikt. Controleer dus in eerste instantie de opdracht.

Er is nog een andere mogelijke oorzaak: u probeert een reservekopie terug te schrijven naar een andere directory dan de directory waar die kopie oorspronkelijk was opgeslagen. Als u een reservekopie hebt gemaakt van een bestand in C:\DATA\TEKSTEN, kunt u die niet terugschrijven naar de directory C:\DATA\BRIEVEN.

Tip: U kunt dit probleem vermijden door op de diskette met de reservekopie het zoekpad te noteren. U kunt ook gebruik maken van de mogelijkheid een protocolbestand aan te leggen. Wat dat is, leggen we uit in de volgende paragraaf.

12.4 Een protocolbestand maken bij een backup

Als u meerdere backups naast elkaar gebruikt om tijd te besparen, raakt u gauw het overzicht kwijt. In welke backup was ook al weer een reservekopie van bestand X gemaakt? MS-DOS biedt u de mogelijkheid de gegevens van een backup bij te houden in een zogenaamd protocolbestand.

Uitwerking

Als u aan de opdracht voor het maken van een backup de optie /L toevoegt, maakt MS-DOS een apart bestand aan, BACKUP.LOG, waarin alle relevante gegevens van de backup worden genoteerd. Wilt u dus bijvoorbeeld een backup maken van de bestanden in de directory C:\DOS en de bijbehorende subdirectories en tevens alle gegevens optekenen in een protocolbestand, dan typt u de volgende opdracht:

```
BACKUP C:\DOS\*.* A: /S /L
```

Print het protocolbestand en bewaar de hardcopy bij de diskettes van de backup, dan weet u altijd wat er op de diskettes staat. U kunt ook een aparte directory C:\LOG aanmaken en daar alle protocolbestanden van backups bewaren. Het voordeel daarvan zullen we hieronder bespreken.

Een protocolbestand van een backup ziet er ongeveer als volgt uit:

```
27-10-1991  20:47:39
001  \DOS\EGA.SYS
```

```
001   \DOS\FORMAT.COM
001   \DOS\NLSFUNC.EXE
001   \DOS\COUNTRY.SYS
001   \DOS\DISPLAY.SYS
001   \DOS\EGA.CPI
001   \DOS\KEYB.COM
001   \DOS\KEYBOARD.SYS
001   \DOS\MODE.COM
001   \DOS\SETVER.EXE
001   \DOS\ANSI.SYS

...
```

Boven het bestand plaatst DOS de datum en het tijdstip van de backup. Vervolgens worden alle opgeslagen bestanden opgesomd, inclusief zoekpad. Het nummer voor de bestandsnamen is het nummer van de backup-diskette.

Maakt u nu een tweede backup, dan koppelt MS-DOS de backup-gegevens daarvan aan het eind van dit eerste protocolbestand. Om dit te voorkomen kunt u de naam van het eerste protocolbestand wijzigen of een nieuwe naam opgeven achter de optie /L, bijvoorbeeld:

```
BACKUP C:\BAT\*.* A: /L BACKBAT.LOG
```

Slaat u alle protocolbestanden op in één directory C:\LOG, dan biedt dat de volgende voordelen:

1 U kunt te allen tijde aan de hand van de datum van het protocolbestand vaststellen, wanneer u van bepaalde bestanden voor het laatst een reservekopie hebt gemaakt.

2 U kunt met behulp van de DOS-opdracht FIND naar een bepaald bestand in uw protocolbestanden zoeken, zodat het u weinig moeite kost de reservekopie te vinden van dat ene bestand dat zojuist verloren is gegaan.

12.5 De datum vastleggen van een backup met XCOPY

Gebruikt u de opdracht XCOPY vaak om een reservekopie te maken van gegevensbestanden, dan is niet zonder meer te achterhalen wanneer u van een bepaald be-

stand voor het laatst een reservekopie hebt gemaakt. Op de diskette met reservekopieën houdt elk bestand de datum waarop het voor het laatst is gewijzigd. Met behulp van een klein batchbestand krijgt u inzicht in deze zaak.

Uitwerking

Houd op de diskette een soort protocolbestandje bij waarin u steeds de actuele datum optekent op het moment dat u met XCOPY bestanden naar de diskette kopieert.

We gaan ervan uit dat u van de bestanden van de directory C:\DATA reservekopieën wilt maken en tegelijkertijd het bestand DATUM.DAT wilt actualiseren. Het onderstaande batchbestand KOPDATA.BAT doet het werk voor u. De kern ervan hebben we in paragraaf 9.4 besproken, waar we de datum opsloegen in een bestand in het kader van beveiliging.

```
@REM KOPDATA.BAT - reservekopie van C:\DATA inclusief actuele datum
@ECHO OFF
XCOPY C:\DATA\*.* A: /M
DATE <C:\BAT\ENTER >A:DATUM.DAT
TIME <C:\BAT\ENTER >>A:DATUM.DAT
@ECHO ON
```

12.6 Een automatische backup op een bepaalde weekdag

Hoe handig de methode ook is waarmee u een backup kunt maken, het helpt u niets als u hem niet gebruikt. De meeste backups worden niet gemaakt, omdat er altijd even geen tijd voor is. Het ligt voor de hand om het maken van een backup op een bepaalde dag van de week dan maar te institutionaliseren.

Uitwerking

Er zijn meer manieren om het doel te bereiken. In deze paragraaf beschrijven we een methode die uitsluitend met batchbestanden werkt. U hebt er dus geen speciale hulpprogramma's voor nodig. Bovendien kunt u bij iedere datum een aantekening maken, die op die bewuste dag automatisch op het scherm verschijnt.

Ook deze batchbestanden horen weer thuis in de directory C:\BAT. Mocht er ergens

iets fout gaan, dan is het van groot belang dat u de uitvoer van de diverse opdrachten naar het scherm nauwkeurig bekijkt, omdat die weer wordt gebruikt als invoer voor andere opdrachten. De kans is groot dat u daar ergens een fout in hebt gemaakt. Het gaat om de volgende batchbestanden.

1 Het bestand ENTER.BAT (zie paragraaf 9.1) dat bestaat uit een nabootsing van de toetsaanslag Enter en dat wordt gebruikt om de opdracht DATE af te sluiten.

2 Het batchbestand AUTO.BAT dat de uitvoer van de interne DOS-opdracht DATE omleidt naar het bestand DAG.BAT. Vervolgens roept het batchbestand dat tweede batchbestand DAG.BAT op.

```
@REM AUTO.BAT - door Manfred Tornsdorf
@REM Een aantekening zoeken bij een bepaalde dag
@REM Bij iedere weekdag een melding uitvoeren
@DATE <C:\BAT\ENTER >C:\BAT\DAG.BAT
@DAG
```

3 Het speciale batchbestand CURRENT.BAT dat verhindert dat MS-DOS de foutmelding ‘File not found’ op uw scherm zet, als het besturingssysteem probeert het batchbestand DAG.BAT uit te voeren. Dat bestand ziet er namelijk als volgt uit.

```
Current date is Sun 27-10-1991
Enter new date (dd-mm-yy):
```

MS-DOS probeert de eerste acht tekens van dit zogenaamde batchbestand te interpreteren als DOS-opdracht. Het speciale batchbestand CURRENT.BAT verhindert de foutmelding die eigenlijk moet volgen omdat MS-DOS geen opdracht CURRENT kent.

```
@REM CURRENT.BAT - foutmelding voorkomen
@CALL REMARK.BAT %4
@ECHO Druk op de [%3]-toets voor een automatische
@ECHO afhandeling van de opdrachten
@ECHO Afbreken met Ctrl-C
PAUSE >NUL
@%3.BAT
```

4 Deze tamelijk omslachtige procedure met DAG.BAT en CURRENT.BAT is nodig, omdat CURRENT.BAT zowel de dag als de datum als parameters krijgt overgedragen. Aansluitend op het bovenstaande voorbeeld:

```
Current date is Sun 27-10-1991
```

is de derde parameter de weekdag ‘Sun’ en de vierde parameter de datum ‘27-10-1991’. CURRENT.BAT roept nu het batchbestand REMARK.BAT op met de vierde parameter, dus de datum. Dat batchbestand kan daarmee vervolgens naar een datum zoeken in een van tevoren aangemaakt bestand:

```
@REM REMARK.BAT
@ECHO Ik zoek naar aantekeningen bij [%1]
@FIND "%1" C:\BAT\REMARK.TXT
```

5 Vanzelfsprekend dient er in de directory C:\BAT een bestand REMARK.TXT te zijn opgeslagen. Daarin slaat u voor iedere keer dat er een backup gemaakt moet worden een item op met de datum (in het juiste formaat), afgesloten door Enter. Daar tussendoor kunt u aantekeningen en geheugensteuntjes kwijt. Het bestand zou er als volgt kunnen uitzien.

```
30-10-1991: groene afval-container bij de straat zetten

31-10-1991: om 10.30 uur tandarts Schwiekman

01-11-1991: BACKUP maken van C:\DATA
```

6 Na REMARK.BAT roept CURRENT.BAT een tweede batchbestand op: %3.BAT. Dat bestand neemt de naam aan van de weekdag in het bestand DAG.BAT. In het bovenstaande voorbeeld dus SUN.BAT. De weekdag wordt namelijk geïnterpreteerd als de derde parameter bij de pseudo-opdracht CURRENT. U kunt nu dus een batchbestand SUN.BAT aanmaken waarin u opdrachten verwerkt die specifiek voor die weekdag gelden. Datzelfde geldt voor de andere zes batchbestanden: MON.BAT, TUE.BAT, enzovoort. Voor de zondag zou u het batchbestand bijvoorbeeld kunnen beginnen met een ECHO-opdracht:

```
@ECHO Eerst maar eens lekker uitslapen...
```

Zinvoller is de volgende inhoud voor bijvoorbeeld het bestand MON.BAT, waarin u iedere maandag een backup maakt van de directory C:\DATA:

```
@REM Opdrachten voor de MAANDAG
@ECHO Vandaag is het maandag
@ECHO - tijd voor een backup van gegevensbestanden
@ECHO Plaats de backup-diskette in drive A:
@ECHO Doorgaan: spatiebalk;
```

```
@ECHO Afbreken: Ctrl-C
@CALL KOPDATA C:\DATA\*.TXT A: 28-10-1991 /S
```

8 Om de hele procedure iedere dag in werking te stellen, hoeft u nu niet meer te doen dan aan het einde van het bestand AUTOEXEC.BAT de volgende regel toe te voegen:

```
@AUTO
```

Tip: De opdracht AUTO dient daadwerkelijk aan het einde van het bestand AUTOEXEC.BAT te staan, omdat de aan elkaar gekoppelde batchbestanden niet allemaal met de opdracht CALL worden gestart. Dat betekent dat de besturing aan het eind van de hele procedure niet meer terugkeert naar het bestand AUTOEXEC.BAT en dat dus opdrachten die na AUTO volgen, worden genegeerd.

12.7 Bestanden herstellen

Het is niet altijd mogelijk verminkte bestanden eenvoudigweg te vervangen door de reservekopie. Als de actuele verminkte versie duidelijk nieuwer is, heeft het geen zin daarvoor een oude reservekopie in de plaats te gebruiken. In dat geval rest u alleen nog een poging het verminkte bestand te herstellen. Datzelfde geldt als u wordt getroffen door een dubbel ongeluk: niet alleen het bestand op de harde schijf is onleesbaar, maar ook de reservekopie op diskette. MS-DOS biedt weinig mogelijkheden om bestanden te herstellen, maar we zullen er twee bespreken.

Een leesfout op een diskette

U wilt gegevens van een diskette lezen, maar het systeem meldt een fout:

```
General failure reading drive A:
Abort, Retry, Fail?
```

of:

```
Unrecoverable read error on drive A:
```

Als u de gegevens beslist nodig hebt, maar u beschikt niet over een reservekopie, kunt u in eerste instantie het beste proberen de gegevens van de defecte diskette naar een andere diskette te kopiëren en vervolgens de tweede diskette te lezen. Hebt

u geluk, dan kunt u de meeste gegevens van die diskette alsnog lezen.

Tip: We raden u aan de gegevens die op deze manier hebt gered, nauwkeurig te controleren op integriteit. Het is heel goed mogelijk dat de bestanden niet meer voor de volle honderd procent in orde zijn; in tekstbestanden herkent u dat bijvoorbeeld aan het plotseling in de tekst opduiken van bijzondere tekens.

Als de kopieeractie geen resultaat heeft, kan MS-DOS u alleen nog verder helpen met de opdracht RECOVER, die we hieronder zullen bespreken in samenhang met leesfouten op de harde schijf.

Leesfouten op de harde schijf

Kunt u plotseling gegevens van de harde schijf niet meer lezen en beschikt u niet over reservekopieën van de betrokken bestanden, dan kan MS-DOS u alleen verder helpen met de opdracht RECOVER. Deze opdracht probeert van een bestand of van een groep bestanden te redden wat er nog te redden valt.

Let op! RECOVER is een uiterst gevaarlijk stuk gereedschap. Gebruikt u de opdracht verkeerd, dan richt hij meer schade aan dan hij kan herstellen. Maak in elk geval eerst een backup van de harde schijf of diskette die u te lijf wilt gaan met RECOVER. Beschikt u over een programma als de Norton Utilities, gebruik dat dan bij voorkeur in plaats van RECOVER.

Kent u de naam van het bestand dat verminkt is, dan kunt u RECOVER activeren met de bestandsnaam als parameter. RECOVER converteert het bestand in een of meer bestanden met de naam FILE0000.CHK, FILE0001.CHK, enzovoort. Het bestand wordt opgedeeld in leesbare stukken; de niet meer leesbare gedeelten gaan definitief verloren.

Als veel bestanden op de harde schijf onleesbaar zijn geworden en hebt u de gegevens letterlijk koste wat het kost nodig, dan kunt u RECOVER ook gebruiken om de hele harde schijf langs te gaan. RECOVER redt dan wat er te redden valt en slaat alle leesbare gedeelten weer op in bestanden met de naam FILExxxx.CHK, die doorlopend worden genummerd.

Let op! RECOVER kan geen onderscheid maken tussen correcte en defecte bestanden. Alle bestanden worden gekopieerd naar de speciale bestanden FILExxxx.CHK. Al die bestanden bevinden zich na beëindiging van de procedure in de hoofddirectory en zijn volstrekt niet meer te herkennen

aan hun naam. Uw directory-structuur is gewist. Het is dus absoluut noodzakelijk om voor u RECOVER start, reservekopieën te maken van alles wat nog intact is.

Het heeft alleen zin RECOVER los te laten op uw harde schijf, als u de meeste bestanden kunt missen, omdat u er een goede reservekopie van hebt. Uit de geweldige lijst met FILExxxx.CHK-bestanden zult u dan naderhand die paar bestanden moeten zoeken waar het om gaat, de rest kunt u wissen. Gebruik de hulpprogramma's die we in paragraaf 6.8 hebben besproken om aan de hand van de inhoud van de bestanden de oorspronkelijke bestanden te reconstrueren.

Het verloren gaan van gegevens of bestanden op een harde schijf heeft natuurlijk een oorzaak. Hebt u werkelijk zo ingrijpend moeten handelen, dat de hele harde schijf overhoop ligt, maakt u dan van de gelegenheid gebruik om hem, voordat u alle gegevens er weer naar toe kopieert, opnieuw te formatteren. MS-DOS zorgt er dan voor dat defecte sectoren niet meer worden gebruikt.

RECOVER kan geen opgemaakte bestanden redden. Als u bijvoorbeeld probeert een tekst te redden die u in de tekstverwerker WORD op een bepaalde manier hebt opgemaakt en RECOVER deelt de tekst op in een paar bestanden, dan kunt u die delen in WORD niet meer bewerken. Het kan zelfs zijn dat uw computer blijft hangen als u het toch probeert. Datzelfde geldt voor de gegevens van een database die in een bepaald formaat zijn opgemaakt en haast vanzelfsprekend voor programma's met de extensies .EXE en .COM.

Tip: Ten slotte herhalen we nog een keer de tip van het begin van deze paragraaf. Er zijn beduidend betere hulpprogramma's op de markt dan RECOVER. Alles met elkaar is het de moeite waard om te overwegen een dergelijk hulpprogramma aan te schaffen als de calamiteiten van deze paragraaf zich voordoen. De bekendste van deze hulpprogramma's zijn PC TOOLS en de Norton Utilities.

12.8 Reservekopieën maken van systeemgebieden van een schijf

Als de vorige paragraaf u het idee heeft gegeven dat u maar beter niet met RECOVER kunt werken, hebt u de boodschap goed begrepen. Dat wil nog niet zeggen dat u voor de bescherming van uw bestanden aan de goden bent overgeleverd. Er zijn een paar belangrijke voorzorgsmaatregelen die problemen kunnen voorkomen. Daarbij hoort het opslaan van de gegevens van de systeemgebieden van de (harde) schijf.

Uitwerking

Bij een harde schijf kunt u uit die systeemgebieden drie belangrijke vormen van informatie beveiligen. Het gaat om:

- de partitietabel
- de FAT (File Allocation Table)
- gegevens over gewiste bestanden.

Een reservekopie van de partitietabel

De partitietabel bevat primaire gegevens over de indeling van uw harde schijf. Iedere keer dat u daar iets in wijzigt, dient u de gegevens over de partitietabel op te slaan op een aparte diskette. U plaatst een geformatteerde diskette in de drive en typt de opdracht:

```
MIRROR /partn
```

Het programma MIRROR maakt nu een kopie van de partitietabel op de diskette. Raakt de partitietabel op uw harde schijf om de een of andere reden verminkt, dan kunt u met de volgende opdracht de reservekopie kopiëren naar de harde schijf:

```
UNFORMAT /partn
```

Let op! Gebruik de opdracht UNFORMAT /partn echt alleen als de partitietabel van de harde schijf is verminkt.

Een reservekopie van de FAT

Veel vaker dan van de partitietabel dient u een reservekopie te maken van de FAT. In de FAT wordt steeds bijgehouden waar bestanden op de schijf te vinden zijn. Die situatie verandert iedere keer dat u een bestand naar de schijf schrijft. U dient voor elke drive van de harde schijf een apart MIRROR-bestand aan te maken. Voor drive C: van de harde schijf ziet de opdracht er als volgt uit:

```
MIRROR C:
```

Het DOS-programma maakt nu in de hoofddirectory een bestand MIRROR.FIL aan. Raakt de FAT om de een of andere reden verminkt, dan kunt het MIRROR-be-

stand gebruiken om de schade te herstellen. Dat gaat ook weer met UNFORMAT:

```
UNFORMAT C:
```

Let op! Ook deze opdracht moet u met de nodige voorzichtigheid hanteren. Hij wist namelijk ook gegevens: UNFORMAT wist alle bestanden die zijn gewijzigd of nieuw aangemaakt sinds u het bestand MIRROR.FIL aanmaakte. Als dat bestand een week oud is, en u hebt nogal wat werk verricht sinds die tijd, kan het handig zijn om het bestand niet meer te gebruiken, of te proberen eerst voorzieningen te treffen die het verlies van nieuwe bestanden voorkomt (reservekopieën). Erg veel is wat dat betreft afhankelijk van de ernst van het probleem. In sommige gevallen kan een beschadigde FAT ertoe leiden dat u vrijwel helemaal niets meer met uw harde schijf kunt beginnen.

Informatie bijhouden over gewiste bestanden

MIRROR heeft nog een derde functie, namelijk die van het bijhouden van de gegevens van gewiste bestanden in een soort protocolbestand. Het programma wordt dan als TSR (Terminate and Stay Resident) geladen en registreert iedere wishandeling. De gegevens worden opgeslagen in een apart bestand: PCTRACKR.DEL. Van dat bestand is het attribuut system gezet, zodat het normaal gesproken niet zichtbaar is. U installeert MIRROR voor dit doel op de volgende manier:

```
MIRROR C: /td
```

De letter 'd' (Drive) is de naam van de drive die MIRROR in de gaten moet houden. Voer die drive-letter in zonder dubbele punt. U kunt ook meer drives opgeven. Wilt u bijvoorbeeld dat MIRROR de informatie bijhoudt voor de harde schijf C: en het diskettestation a:, dan geeft u de volgende opdracht:

```
MIRROR C: /tc /ta
```

Met de opdracht UNDELETE kunt u nu eenvoudiger per ongeluk gewiste bestanden herstellen. UNDELETE werkt ook zonder MIRROR, maar de kans op succes is groter als het bestand PCTRACKR.DEL wordt bijgehouden. Vanzelfsprekend wordt de kans op succes kleiner, naarmate u na het wissen meer bestanden naar de schijf hebt geschreven: er komt een moment waarop over het gewiste bestand heen wordt geschreven.

Tip: De opdracht die u hierboven hebt ingevoerd, is slechts actief zolang de

computer is ingeschakeld. Na een reset is MIRROR niet automatisch opnieuw actief. Om dat te bereiken neemt u de opdracht op in uw AUTOEXEC.BAT-bestand.

12.9 FORMAT ongedaan maken

Het ergste wat u kan overkomen is per ongeluk de harde schijf formatteren. Tot versie 5.0 waren de gevolgen daarvan rampzalig voor uw bestanden. Gelukkig heeft Microsoft in versie 5.0 de FORMAT-procedure zo aangepast, dat herstel mogelijk is.

Uitwerking

Bij die herstelprocedure hoort de nieuwe DOS-opdracht UNFORMAT. Om bijvoorbeeld een per ongeluk geformatteerde schijf in drive A: te herstellen in de oude toestand, typt u de opdracht:

```
UNFORMAT A:
```

Let op! Hebt u die schijf na de opdracht FORMAT nog weer gebruikt om een of meerdere bestanden naar toe te schrijven, dan is succes twijfelachtig. In de eerste plaats gaan die nieuwe bestanden verloren, maar daarnaast zijn de oude gegevens door die schrijfhandelingen verstoord, zodat UNFORMAT er wellicht niets meer van kan maken. Ook deze opdracht dient u dus met de nodige voorzichtigheid te hanteren.

Als u de schijf opnieuw hebt geformatteerd met de optie /U, dan zijn de gegevens niet meer te redden. De optie zorgt er namelijk voor dat oude gegevens definitief worden gewist. Ook dient u te beschikken over het bestand MIRROR.FIL om het herstel tot een goed einde te brengen.

UNFORMAT in combinatie met MIRROR

Als met behulp van MIRROR reservekopieën van de systeemgebieden hebt gemaakt, is de kans aanzienlijk groter dat UNFORMAT de schade kan herstellen.

Wilt u de schade herstellen van een per ongeluk geformatteerde harde schijf C:, gebruikt u de opdracht:

```
UNFORMAT C:
```

UNFORMAT zoekt nu uit of er een bestand MIRROR.FIL op de harde schijf staat en of het dat kan gebruiken. Omdat bij het formatteren met FORMAT automatisch een dergelijk bestand wordt aangemaakt, dat dan natuurlijk nog leeg is, vraagt UNFORMAT u of u gebruik wilt maken van dit bestand of van het bestand MIRROR.BAK. Gelukkig wordt namelijk vóór het aanmaken van een nieuw bestand MIRROR.FIL het bestaande bestand eerst hernoemd tot MIRROR.BAK. U dient op te geven of u de nieuwste (Last) versie wilt gebruiken of de vorige (Prior). Druk op de geaccentueerde toets. Bij verstek gebruikt UNFORMAT de nieuwste versie, wat als het slechts om een enkel bestand gaat, ook heel verstandig is.

Let op! Een verkeerd gebruik van UNFORMAT kan recente gegevens verminken en dus nog veel grotere schade aanrichten. U kunt de procedure met Esc afbreken en u eerst nog eens goed afvragen wat de consequenties zijn. Gebruikt u UNFORMAT voor een diskette, maak dan altijd eerst een reservekopie van die diskette met bijvoorbeeld DISKCOPY.

UNFORMAT zonder MIRROR

Beschikt u niet over met MIRROR aangemaakte kopieën van de systeemgebieden van de schijf, dan kan UNFORMAT wellicht de schade toch nog herstellen. U dient dan de optie /U aan de opdracht toe te voegen. De kans op succes wordt wel kleiner:

```
UNFORMAT A: /U
```

Na een controlevraag doorzoekt UNFORMAT naar bestaande gegevens (bestanden en directories). Pas na een tweede controlevraag brengt UNFORMAT vervolgens wijzigingen aan op de schijf.

Tip: Om eerst een keer het resultaat van de procedure te bekijken kunt u UNFORMAT starten met de optie /TEST. UNFORMATE verricht dan alle handelingen, met uitzondering van het wijzigen van de gegevens op de schijf, zodat er nog geen gevaar bestaat dat u gegevens onherstelbaar beschadigt.

12.10 De inhoud van BACKUP-diskettes achterhalen

U hebt enige tijd geleden een backup gemaakt van een deel van de harde schijf zonder daaraan een protocolbestand te koppelen of anderszins te noteren wat er op de diskettes staat. Met behulp van een speciaal programma LISTBACK, kunt u echter

de inhoud lezen van het control-bestand van backup-diskettes.

Uitwerking

Iedere diskette van een backup bevat vanaf MS-DOS 3.3 twee bestanden:

```
 Volume in drive A is BACKUP   001
 Volume Serial Number is 1E0C-7646
 Directory of A:\

BACKUP   001     726016 27-10-91   20:49
CONTROL  001       1331 27-10-91   20:49
        2 file(s)     727347 bytes
                        2048 bytes free
```

Het eerste bestand BACKUP.001 bevat de werkelijke gegevens van de backup. Het tweede bestand CONTROL.001 bevat belangrijke informatie over directories en bestanden van die gegevens. Het nummer in de extensie van beide bestanden is steeds het nummer van de backup-diskette.

Als u probeert met de opdracht

```
TYPE A:CONTROL.001
```

de inhoud van dat bestand te bekijken, verschijnt er een prachtig mozaïek van tekentjes op uw scherm met daar tussendoor hier en daar een directory- of bestandsnaam. Alle informatie staat er wel, maar is kennelijk niet in een leesbare vorm opgeslagen. Met het onderstaande programma LISTBACK.BAS wordt de inhoud van het control-bestand vertaald.

U kunt als u het programma gebruikt, een willekeurige diskette in de drive plaatsen om te kijken wat er in het control-bestand staat. Dat is in tegenstelling met de opdracht RESTORE van MS-DOS, die alleen werkt als u de diskettes in de juiste volgorde in de drive plaatst. U dient wel bij het starten van het programma aan te geven wat het nummer van de backup-diskette is. Dat nummer achterhaalt u eenvoudig met de opdracht DIR.

```
CLS
PRINT "*************************************************************"
PRINT "**   Programma voor het bekijken van control-bestanden     **"
PRINT "**                van backup-diskettes                     **"
```

```
PRINT "********************************************************"
PRINT "Druk op een toets..."
Start:
INPUT "Wat is het nummer van de diskette"; DiskNr
IF DiskNr < 1 or DiskNr > 999 THEN PRINT "Ongeldig nummer (1-999)":
                                   GOTO Start
DiskNr$ = STR$(DiskNr)
DiskNr$ = MID$(DiskNr$, 2, LEN(DiskNr$) - 1)
WHILE LEN(diskNr$) < 3
  DiskNr$ = "0" + DiskNr$
WEND
ControlBestand$ = "A:\CONTROL." + DiskNr$

REM Controleren of de juiste schijf in de drive is geplaatst
ON ERROR GOTO Fout
OPEN ControlBestand$ FOR INPUT AS #1
CLOSE #1
ON ERROR GOTO 0
OPEN ControlBestand$ FOR BINARY AS #1
REM *********Header van het Control-bestand lezen************
DoDisk:
Aan = 1: REM lengtebyte van de header lezen
GOSUB Lezen
PRINT "Lengte van de header = "; HEX$(Na)
Headerlen = Na
Aan = 8: REM naam lezen
GOSUB Lezen
PRINT "Naam = "; N$
Aan = 1: REM Diskettenummer lezen
GOSUB Lezen
PRINT "Diskettenummer van de backup = "; Na
Aan = Headerlen - 10: REM restant Header hernieuwd lezen
GOSUB Lezen
REM *********Gegevensstring lezen*****************************
GetBlock:
WHILE INKEY$ = "": WEND
PRINT
Aan = 1: REM lengte lezen (70 = directory, 34 = bestand)
GOSUB Lezen
BlockLen = Na : REM lengte vastleggen
IF BlockLen = 0 THEN GOTO Einde
IF Na = 70 THEN
```

```
  PRINT "Directory: "; : REM **************** directory
  Aan = 63 : REM directory-naam lezen
  GOSUB Lezen
  DirNaam$ = N$
  PRINT DirNaam$ : REM naam uitvoeren
  Aan = 1 : REM lowbyte AanBst lezen
  GOSUB Lezen
  AanBst = Na
  Aan = 1 : REM hibyte lezen
  GOSUB Lezen
  AanBst = AanBst + 256 * Na
  PRINT "Aantal bestanden in deze directory op deze schijf: ";
                                                               AanBst
  Aan = 4 : REM restbytes lezen (adres volgende directory of FF FF)
  GOSUB Lezen
  GOTO GetBlock
ELSE
IF Na = 34 THEN
  PRINT "Bestand: "; : REM ***************** bestand
  Aan = 12 : REM bestandsnaam lezen
  GOSUB Lezen
  PRINT N$
  Aan = 1 : REM Split-byte (3 = volledig bestand, 2 = rest volgt)
  GOSUB Lezen
  IF Na = 2 THEN PRINT " Bestand niet volledig, restant op volgende
                                                              schijf"
  DatLen = 0
  Faktor = 1
  FOR i = 1 TO 4
    Aan = 1
    GOSUB Lezen
    DatLen = DatLen + Na * Faktor
    Faktor = Faktor * 256
  NEXT i
  PRINT "Bestandslengte: "; DatLen
  Aan = 2 : REM twee bytes (01,00) lezen
  GOSUB Lezen
  DatStart = 0
  Faktor = 1
  FOR i = 1 TO 4
    Aan = 1
    GOSUB Lezen
```

```
      DatStart = DatStart + Na * Faktor
      Faktor = Faktor * 256
    NEXT i
    PRINT "Startadres van het bestand: "; DatStart
    DatLenDisk = 0
    Faktor = 1
    FOR i = 1 TO 4
      Aan = 1
      GOSUB Lezen
      DatLenDisk = DatLenDisk + Na * Faktor
      Faktor = Faktor * 256
    NEXT i
    PRINT "Bestandslengte op deze diskette: "; DatLenDisk

    Aan = 6 : REM attributen, datum en dergelijke lezen
    GOSUB Lezen
    AanBst = AanBst - 1
    IF AanBst = 0 THEN
      PRINT "-------------------"
    END IF
    GOTO GetBlock
  ELSE
    PRINT "Fout: ongeldige bloklengte: "; Na
  END IF: REM if 34
  END IF: REM if 70 or 34
  Einde:
  PRINT "Einde programma"
  CLOSE #1
  END

  Lezen:
  N4 = INPUT$(Aan, #1)
  Na = ASC(N4 + CHR$(0))
  RETURN

  Fout:
  BEEP
  PRINT "Plaats de juiste diskette in de drive en druk op een toets."
  PRINT "Druk op E (=Einde) om het programma af te breken."

  Wacht:
  Invoer$ = INKEY$
```

```
IF Invoer$ = "" THEN GOTO Wacht
IF Invoer$ = "e" OR Invoer$ = "E" THEN CLOSE : END
RESUME
```

Nadat u het programma hebt gestart, typt u het nummer van de backup-diskette. Als u bijvoorbeeld de eerste diskette van de backup wilt lezen, dan typt u '1', en verschijnt het volgende op uw scherm:

```
Lengte van de header = 8B
Naam = BACKUP
Diskettenummer van de backup = 1
```

U drukt op een toets en de inhoud van de eerste directory verschijnt op uw scherm, bijvoorbeeld:

```
Directory: DOS
Aantal bestanden in deze directory op deze schijf: 5
```

U drukt opnieuw op een toets en het programma geeft u de gegevens van het eerste bestand:

```
Bestand: ANSI.SYS
Bestandslengte: 1647
Startadres van het bestand: 0
Bestandslengte op deze diskette: 1647
```

Na de naam van het bestand volgt de grootte ervan. Dan het startadres op de backup-diskette en ten slotte dat deel van het bestand dat op deze diskette staat: in dit geval alles. Als het laatste bestand van een diskette niet in zijn geheel op de diskette staat, volgt de boodschap:

```
Bestand niet volledig, restant op volgende schijf
```

Als alle directories en bestanden van een diskette zijn weergegeven, wordt het programma afgesloten. U kunt het programma dan opnieuw starten om naar de gegevens van een andere diskette te kijken.

Tip: Als het programma om de een of andere reden blijft steken, drukt u op Enter om verder te gaan.

Toelichting

Een control-bestand van BACKUP bestaat uit een *Header* en de afzonderlijke blokken met directories of bestanden. De header is als volgt opgebouwd.

nummer	voorbeeld	inhoud
01	139	grootte van de header inclusief deze byte (139 = $8B)
02-09		acht bytes voor namen (BACKUP, aangevuld met spaties)
10	01	nummer van de diskette
11-138	00	niet in gebruik, bevat nullen
139	00	Flag die aangeeft of er nog meer diskettes volgen:
		00 wil zeggen dat er nog meer diskettes zijn;
		255 (=$FF) geeft de laatste diskette aan.

Daarop volgen de gegevens over de directories en bestanden. Een blok waarin een directory wordt beschreven, ziet er als volgt uit.

nummer	voorbeeld	inhoud
1	70	grootte van het blok inclusief deze byte (70 = $46)
2-46		naam van de directory (zoekpad)
65/66	09 00	het aantal subdirectories van deze directory
		(in de vorm van een word met lowbyte en een high-byte)
67/68	BF 01	het absolute adres van de volgende directory;
		$FF $FF geeft aan dat er geen volgende directory is
69/70	FF FF	FFFF of 0000

Dan volgt het blok met het eerste bestand:

nummer	voorbeeld	inhoud
01	34	grootte inclusief deze byte $22
02-13		twaalf tekens voor de naam, niet gebruikte plaatsen worden opgevuld met \0
14	3	Split-byte; de waarde drie geeft aan dat het bestand compleet is,
		de waarde twee dat er een restant is opgeslagen op een volgende diskette
15-18	6F 06 0 0	grootte van het bestand in de vorm van een Longint,
		dat wil zeggen twee words, eerst low, dan high en binnen de words low/high.
		Voor het voorbeeld dus: $0000066F = 1647
19	01	?
20	00	?
21-24	0 0 0 0	startadres in het backup-bestand (longinteger)
25-28	6F 06 0 0	lengte van het bestand op deze diskette
29	0	attribuut-byte
30	00	?
31-34	5C 7D 05 00	datum en tijdstip

Na de header komt altijd een blok met de directory voor de daarna volgende bestanden. Dat gebeurt ook als er op de vorige diskette van de backup al bestanden uit die directory staan. Dat is precies de reden waarom u met het programma een willekeurige diskette kunt bekijken.

Als alle bestanden van een directory verwerkt zijn en er volgt een nieuwe directory, volgt dus eerst weer een directory-blok. RESTORE en LISTBACK herkennen zo'n blok aan zijn lengte: 70 bytes in plaats van 34 bytes.

12.11 Bestanden herstellen zonder RESTORE

Het ergste wat u kan overkomen bij het teruglezen van een backup is een beschadigde diskette. Veronderstel dat u backup hebt met 29 diskettes en u start RESTORE. Bij de zevende diskette blijkt RESTORE het control-bestand niet te kunnen lezen: RESTORE breekt af en u kunt de overige bestanden ook niet meer teruglezen. Dat maakt RESTORE nogal onbetrouwbaar. We hebben daarom een programma geschreven in QBASIC dat het probleem voor u kan oplossen: GETBACK.BAS.

Uitwerking

GETBACK.BAS kan een backup lezen en terugschrijven naar de harde schijf vanaf een willekeurig gekozen diskette. U kunt bovendien bij ieder bestand aangeven of het wel of niet wilt terugschrijven.

```
REM Programma voor het lezen van bestanden van BACKUP 3.3
CLS
PRINT "***********************************************************
PRINT "** Programma voor het lezen van bestanden van BACKUP 3.3 *
PRINT "**                 door Manfred Tornsdorf                *
PRINT "**           Gebruik het programma zonder APPEND         *
PRINT "***********************************************************

doerrornr = 0 : REM foutnummer voor bijzondere routines
bstrest = 0 : REM bestand niet volledig, restant op volgende schijf
bstopen = 0 : REM het bestand is geopend om te schrijven
moredisk = 0 : REM Flag voor meer diskettes = "J"
alle = 0 : niet alle bestanden automatisch terugschrijven

ON ERROR GOTO fout
start:
INPUT "Wilt u het diskettenummer invoeren"; disknr
IF disknr < 1 OR disknr > 999 THEN PRINT "Ongeldig nummer (1-999):
                                                            GOTO start

donewdisk:
GOSUB getcontrolname
GOSUB testdisknummer
OPEN controlbst$ FOR BINARY AS #1
OPEN backupbst$ FOR BINARY AS #2
```

```
dodisk:
GOSUB readheader
REM ***************** gegevensstring lezen ********************
getblock:
PRINT
GOSUB getblocklen

IF blocklen = 0 THEN GOTO diskklaar
REM uitzoeken of er meer diskettes zijn
IF blocklen = 70 THEN GOSUB getdir : GOTO getblock : REM directory
IF blocklen = 34 THEN
REM bestand
GOSUB getdat
IF alle = 1 THEN invoer$ = "a" : GOTO allemaal
IF bstopen = 1 THEN invoer$ = "J" : GOTO allemaal
REM alleen als geen bestand geopend is om te schrijven, vragen
INPUT "bestand terugschrijven (J/N) (E - Einde) (A - Allemaal)";
                                                          invoer$
allemaal:
IF invoer$ = "E" OR invoer$ = "e" THEN GOTO einde
IF invoer$ = "J" OR invoer$ = "j" OR IF invoer$ = "A" OR invoer$
                                                       = "a" THEN
  IF invoer$ = "A" OR invoer$ = "a" THEN alle = 1
  GOSUB restorebst
END IF : bestand teruglezen
GOTO getblock
END IF : REM bestand
PRINT "fout: ongeldige bloklengte: "; blocklen : CLOSE : END

diskklaar:
IF moredisk = 1 THEN
  REM Er zijn nog meer diskettes
  CLOSE #1 : REM CONTROL.??? sluiten
  CLOSE #2 : REM BACKUP.??? sluiten
  disknr = disknr + 1 : REM Diskettenummer met 1 ophogen
  BEEP
  PRINT "Plaats diskette "; disknr; " in de drive"
  PRINT "en druk op Enter."
  WHILE INKEY$ = "" : WEND
  GOTO donewdisk
END IF : REM Er zijn nog meer diskettes
```

```
Einde:
PRINT "Einde programma"
CLOSE
END

REM ******************SUBROUTINES************************
lezen:
n$ = INPUT$(aan, #1)
na = ASC(n$ + CHR$(0))
RETURN

getcontrolname:
disknr$ = STR$(disknr)
disknr$ = MID$(disknr$, 2, LEN(disknr$) - 1)
WHILE LEN(disknr$) < 3
  disknr$ = "0" + disknr$
WEND
controlbst$ = "a:\CONTROL." + disknr$
BACKUPBST$ = "a:\backup." + disknr$
RETURN

readheader
REM ***********header van het control-bestand lezen *********
aan = 1 : REM lengte-byte van de header lezen
GOSUB Lezen
PRINT "lengte van de header = "; HEX$(na)
headerlen = na
IF headerlen <> 139 THEN
  BEEP
  PRINT "Ongeldige header control-bestand"
  CLOSE
  GOTO einde
END IF
aan = 8: REM naam lezen
GOSUB lezen
PRINT "naam = "; n$
aan = 1: REM diskettenummer lezen
GOSUB lezen
disknummer = na
PRINT "diskettenummer van de backup = "; disknummer
IF disknummer <> disknr THEN
  BEEP
```

```
  PRINT "Ongeldig diskettenummer"
  GOTO donewdisk
END IF
aan = headerlen - 11: REM restant header hernieuwd lezen
GOSUB lezen
aan = 1 : REM flag lezen voor de volgende diskette
GOSUB lezen
IF na = 0 THEN moredisk = 1
IF na = 255 THEN moredisk = 0
PRINT "volgende diskette = ": moredisk
RETURN

getblocklen:
aan = 1: REM lengte lezen (70 = directory, 34 = bestand)
GOSUB lezen
blocklen = na : REM lengte vastleggen
RETURN

getdir:
PRINT "directory: "; : REM **************** directory
aan = 63 : REM directory-naam lezen
GOSUB lezen
naamoud$ = n$
GOSUB getnaam : REM nullen verwijderen
dirnaam$ = "\" + naamnieuw$
PRINT dirnaam$ : REM naam uitvoeren

aan = 1 : REM lowbyte aanbst lezen
GOSUB lezen
aanbst = na
aan = 1 : REM hibyte lezen
GOSUB lezen
aanbst = aanbst + 256 * na
PRINT "Aantal bestanden in deze directory op deze schijf: "; aanbst
Aan = 4 : REM restbytes lezen (adres volgende directory of FF FF)
GOSUB lezen
GOSUB isdir : REM controleren of er een volgende directory is
RETURN

getdat:
PRINT "bestand: "; : REM **************** bestand
aan = 12 : REM bestandsnaam lezen
```

```
GOSUB lezen
naamoud$ = n$
GOSUB getnaam : nullen verwijderen
bstnaam$ = naamnieuw$
bestandsnaam$ = dirnaam$ + "\" + bstnaam$
PRINT "<" + bestandsnaam$ + ">"
aan = 1 : REM Split-byte (3 = volledig bestand, 2 = rest volgt)
GOSUB lezen
IF na = 2 THEN
  PRINT " Bestand niet volledig, restant op volgende schijf"
  bstrest = 1 : REM situatie vastleggen
ELSE
  bstrest = 0 : REM bestand compleet op deze diskette
END IF

datlen = 0
faktor = 1
FOR i = 1 TO 4
  aan = 1
  GOSUB lezen
  datlen = datlen + na * faktor
  faktor = faktor * 256
NEXT i
PRINT "bestandslengte: "; datlen

aan = 2 : REM twee bytes (01,00) lezen
GOSUB lezen

datstart = 0
faktor = 1
FOR i = 1 TO 4
  aan = 1
  GOSUB lezen
  datstart = datstart + na * faktor
  faktor = faktor * 256
NEXT i
PRINT "startadres van het bestand: "; datstart

datlendisk = 0
faktor = 1
FOR i = 1 TO 4
  aan = 1
```

```
  GOSUB lezen
  datlendisk = datlendisk + na * faktor
  faktor = faktor * 256
NEXT i
PRINT "bestandslengte op deze diskette: "; datlendisk

aan = 6 : REM attributen, datum en dergelijke lezen
GOSUB lezen
aandat = aandat - 1
IF aandat = 0 THEN PRINT "-------------------"
RETURN

getnaam: REM naamoud$, naamnieuw$
naamnieuw$ = ""
naamlen = LEN(naamoud$)
i = 1
WHILE (i <= naamlen)
  teken$ = MID$(naamoud$, i, 1)
  IF ASC(teken$) = 0 THEN RETURN
  naamnieuw$ = naamnieuw$ + teken$
  i = i + 1
WEND
RETURN

testdisknummer:
doerrornr = 53  : REM foutnummer voor 'bestand niet gevonden'
statusbst = 1  : REM ervan uitgaan dat het bestand er is
OPEN controlbst$ FOR INPUT AS #1
CLOSE #1
IF statusbst = 0 THEN
  BEEP
  PRINT "plaats de juiste diskette in de drive"
  PRINT "en druk op een toets (E = einde)"
  wacht:
  invoer$ = INKEY$
  IF invoer$ = "" THEN GOTO wacht
  IF invoer$ = "e" OR invoer$ = "E" THEN CLOSE : END
  GOTO testdisknummer
END IF
doerrornr = 0  : REM foutroutine uitschakelen
statusbst = 0
RETURN
```

```
fout: REM foutroutine
IF doerrornr = ERR THEN
  REM bijzondere foutroutine vereist
  IF doerrornr = 76 THEN GOTO geendir
  IF doerrornr = 53 THEN GOTO geenbst
END IF
PRINT "fout        : "; ERR
PRINT "in regel    : "; ERL
PRINT "interne code: "; ERDEV
PRINT "device      : "; ERDEV$
foutreactie:
BEEP
INPUT "Abort, retry, fail"; invoer$
IF invoer$ = "a" OR invoer$ = "A" THEN CLOSE : END
IF invoer$ = "r" OR invoer$ = "R" THEN RESUME
IF invoer$ = "f" OR invoer$ = "F" THEN RESUME NEXT
GOTO foutreactie

isdir: REM controleren of een directory bestaat
doerrornr = 76 : REM foutnummer voor 'directory niet gevonden'
nieuwdir$ = ""
i = 2
WHILE i <= LEN(dirnaam$)
  IF MID$(dirnaam$, i, 1) = "\" THEN
    nieuwdir$ = MID$(dirnaam$, 1, i - 1)
    CHDIR nieuwdir$
  END IF
  i = i + 1
WEND
nieuwdir$ = MID$(dirnaam$, 1, i)
CHDIR nieuwdir$
doerrornr = 0 : REM foutroutine uitschakelen
RETURN

restorebst:
IF bstopen = 0 THEN
  doerrornr = 53 : REM foutnummer voor 'bestand niet gevonden'
  statusbst = 1 : REM ervan uitgaan dat het bestand er is
  OPEN bstnaam$ FOR INPUT AS #3
  CLOSE #3
  doerrornr = 0 : REM foutroutine uitschakelen
```

```
  IF statusbst = 1 THEN
    INPUT "bestand gevonden -vervangen (J/N)"; invoer$
    IF invoer$ <> "j" AND invoer$ <> "J" THEN RETURN
    REM bstrest nog regelen!
  END IF :REM statusbst = 1
  OPEN bstnaam$ FOR OUTPUT AS #3
END IF : REM bestand niet geopend

IF datstart <> 0 THEN SEEK #2, datstart + 1
REM pointer op BACKUP zetten
WHILE i + 4096 < datlendisk
  invoer$ = INPUT$(4096, #2)
  PRINT #3, invoer$;
  i = i + 4096
  PRINT ".";
WEND
invoer$ = INPUT$(datlendisk - I + 1, #2)
PRINT #3, invoer$
PRINT "OK"

IF bstrest = 0 THEN
  CLOSE #3 : REM bestand is compleet op deze diskette
  bstopen = 0 : REM situatie vastleggen
ELSE
  bstopen = 1 : REM restant bestand op volgende diskette
END IF
RETURN

REM doerror-routines
geendir : REM directory niet gevonden
PRINT "directory <"; nieuwdir$; "> is niet gevonden.
                                                Aanmaken (J/N)?";
IF alle = 1 THEN invoer$ = "j" : GOTO invoer1
REM alle bestanden automatisch terugschrijven
INPUT invoer$
invoer1:
IF invoer$ = "j" OR invoer$ = "J" THEN
  MKDIR nieuwdir$
  CHDIR nieuwdir$
  RESUME
END IF
```

```
PRINT "Einde programma" : CLOSE : END

geenbst:
REM fout bestand niet gevonden ->
REM flag voor hoofdroutine terugzetten
statusbst = 0 : bestand niet gevonden
RESUME NEXT
```

Nadat u het programma hebt gestart, voert u het nummer van de diskette in. Dat nummer kunt u snel achterhalen met de opdracht DIR, omdat het de extensie is van de bestandsnamen op de diskette. GETBACK geeft u dan vervolgens informatie over de eerste gevonden directory met het eerste daarin gevonden bestand.

U kunt dan kiezen of u dat bestand wilt terugschrijven (j) of overslaan (n). Met E (einde) geeft u aan dat u GETBACK wilt afbreken en met A (alle) dat u alle bestanden zonder verdere controlevragen wilt terugschrijven. Als een bestand al bestaat op de harde schijf stelt het programma overigens altijd een controlevraag, ook als u hebt gekozen voor automatisch terugschrijven van alle bestanden.

Als de directory van het terug te schrijven bestand niet wordt gevonden, vraagt het programma u of het die directory aan moet maken.

Tip: Gebruik het programma niet als u met behulp van de DOS-opdracht APPEND een zoekpad voor gegevensbestanden hebt gespecificeerd. Als namelijk een gelijknamig bestand voorkomt in het zoekpad, zal DOS dat bestand vervangen door het bestand van de backup. U wist het zoekpad met opdracht:

```
APPEND
```

Pas daarna start u GETBACK.

Toelichting

GETBACK haalt de gegevens die nodig zijn voor het terugschrijven van de bestanden evenals LISTBACK uit het control-bestand. Daarnaast moeten er nog wel wat problemen worden opgelost.

Het is heel goed mogelijk dat niet alleen de directory waaruit het terug te schrijven bestand afkomstig is, niet meer bestaat, maar dat ook de directories daarboven niet meer bestaan. Als bijvoorbeeld een bestand BOEK.TXT in de directory C:\TEKS-

TEN\PRIVE moet worden teruggeschreven en zelfs de directory C:\TEKSTEN niet meer bestaat, kan het programma niet zonder meer een directory C:\TEKSTEN\PRIVE aanleggen.

De subroutine *isdir* controleert daarom de volledige padnaam van links naar rechts om na te gaan of alle directories aanwezig zijn. Als dat niet zo is, springt het programma naar een foutroutine die u vraagt of die directory weer moet worden aangemaakt. Reageert u met 'N', dan breekt het programma af, omdat dan de bestanden niet kunnen worden teruggeschreven. Reageert u met 'J', dan wordt de directory aangemaakt en de controle voortgezet.

Een ander probleem is dat van bestanden die al op de harde schijf aanwezig zijn. Daar mag niet zonder meer overheen geschreven worden. De subroutine *restorebst* zet de variabele statusbst op 1 en opent het bestand. Als het bestand niet wordt gevonden, springt het programma automatisch naar een foutroutine, die de variabele statusbst weer op 0 zet. Restorebst stelt aan de hand daarvan vast of het bestand op de harde schijf voorkomt en onderneemt vervolgens actie. Is de waarde van statusbst nog steeds 1 (het bestand is dus aanwezig, want het kan worden geopend), dan vraagt het programma u of er overheen moet worden geschreven.

De DOS-opdracht APPEND voorkomt een goed functioneren van restorebst. Als er namelijk in het zoekpad van APPEND een gelijknamig bestand voorkomt, wordt geen fout gemeld, ook al ontbreekt het bestand in de directory die op dat moment wordt bewerkt. In plaats daarvan schrijft het programma het bestand over dat andere bestand, dat is gevonden via het zoekpad van APPEND.

Het derde probleem betreft het terugschrijven van grote bestanden die verdeeld zijn over meerdere diskettes. De *split-byte* van dergelijke bestanden heeft in dat geval niet de waarde drie, maar twee. Als de subroutine *getdat* die waarde tegenkomt, wordt de variabele restbst op 1 gezet; als een bestand in zijn geheel op één diskette staat, heeft restbst de waarde 0. De subroutine restorebst sluit het bestand nu niet af als alle gegevens van een diskette gelezen zijn, maar gaat na het verwisselen van de diskettes verder met schrijven.

GETBACK onderschept fouten die kunnen optreden en meldt die met eigen foutmeldingen. Vervolgens kunt u aangeven of u het programma wilt afbreken, dan wel de betreffende routine wilt overslaan of herhalen. Slaat u een routine over, dan loopt u het risico, dat het bestand niet correct wordt teruggeschreven. Onder bepaalde omstandigheden kan het echter van belang zijn te redden wat er te redden valt, bijvoorbeeld als van een groot bestand een paar sectoren beschadigd zijn.

Het foutnummer dat GETBACK vermeldt bij een fcutmelding, kunt u opzoeken in

het handboek van QBASIC of QUICKBASIC. U vindt dan een korte beschrijving en meestal een aanwijzing om het probleem op te lossen.

13 Quicktips

In dit hoofdstuk beschrijven we een groot aantal tips, die u los van elkaar kunt toepassen om het maximale uit uw computer te halen onder MS-DOS 5.0.

13.1 Help-teksten verzamelen in een bestand

Met het volgende kleine batchbestand VERZHELP.BAT verzamelt u alle help-teksten bij externe DOS-opdrachten in een bestand.

```
@REM help-teksten verzamelen in een bestand HELP.TXT
@ECHO OFF
IF EXIST HELP.TXT DEL HELP.TXT
FOR %%a IN (*.EXE *.COM) DO %%A /? >>HELP.TXT
```

Start het batchbestand met de opdracht:

```
VERZHELP C:\DOS
```

13.2 COMMAND.COM altijd beschikbaar

Beschikt u niet over een harde schijf, dan is het nogal storend dat het besturingssysteem u voortdurend opdraagt de diskette met COMMAND.COM in drive A: te plaatsen. U lost het probleem op met een kleine RAM-disk, waarop alleen COMMAND.COM staat. Met behulp van de variabele COMSPEC geeft u DOS aan dat het besturingssysteem daar te vinden is. Voeg aan het bestand CONFIG.SYS de volgende regel toe:

```
DEVICE = RAMDRIVE.SYS 50
```

Het bestand RAMDRIVE.SYS dient in de hoofddirectory van de systeemdiskette waarmee u de PC start, te staan en RAMDRIVE.SYS dient de letter C: te gebruiken voor de RAM-disk. De drive-letter wordt gemeld bij de startprocedure. Werkt u niet met MS-DOS, maar met PC-DOS dan gebruikt u niet RAMDRIVE.SYS, maar VDISK.SYS. Aan het bestand AUTOEXEC.BAT voegt u de volgende regels toe:

```
COPY COMMAND.COM C:\
SET COMSPEC = C:\COMMAND.COM
```

Het bestand COMMAND.COM is nu resident geladen en staat voortdurend ter beschikking.

13.3 Programmabestanden verbergen

Als u uw programmabestanden tot op zekere hoogte wilt beveiligen, zodat niet iedere gebruiker ze zonder meer kan starten, kunt u ze verbergen: u verwijdert de extensie .EXE of .COM en start ze daarna met behulp van een apart batchbestand. Wijzigt u bijvoorbeeld de naam van het programma FORMAT.COM in FORMAT, dan kan een argeloze gebruiker er geen diskettes meer mee formatteren.

Het batchbestand DOCOM.BAT hangt vervolgens de juiste extensie aan de bestandsnaam, zodat het weer bruikbaar wordt. Het bestand ziet er als volgt uit:

```
@REM ****************************************************
@REM * batchbestand voor het starten van .COM-bestanden *
@REM ****************************************************
@ECHO OFF
IF "%1" == "" GOTO einde
RENAME %1 COM.COM
COM %2 %3 %4 %5
RENAME COM.COM %1
:einde
@ECHO ON
```

Op vergelijkbare manier maakt u een batchbestand DOEXE.BAT, waarin u overal 'COM' vervangt door 'EXE'.

Tip: Het programma werkt alleen als het bestand met de gewijzigde naam zich in de actuele directory bevindt. Normaal gesproken doorzoekt MS-DOS bij het starten van een programma alle directories van het zoekpad, maar na de opdracht RENAME gebeurt dit niet.

13.4 Het aantal bestanden in de hoofddirectory beperken

Wellicht hebt u zo veel bestanden in de hoofddirectory opgeslagen, dat het geheel volstrekt onoverzichtelijk is geworden. En dat terwijl voor het functioneren van de PC alleen de bestanden AUTOEXEC.BAT en CONFIG.SYS in de hoofddirectory hoeven te staan. Allerlei drivers (*.SYS) en ook COMMAND.COM kunt u echter

zonder problemen in een subdirectory opslaan. U gaat daartoe als volgt te werk:

Device drivers verplaatsen

U brengt alle drivers onder in de directory C:\DOS met de opdracht:

```
COPY C:\*.SYS C:\DOS
```

Vervolgens verwijdert u de bestanden uit de hoofddirectory met de opdracht:

```
DEL C:\*.SYS
```

Omdat het bestand CONFIG.SYS wel in de hoofddirectory dient te staan, kopieert u dat weer terug met de opdracht:

```
COPY C:\DOS\CONFIG.SYS C:\
```

Vervolgens past u het zoekpad in het bestand CONFIG.SYS aan. Daar staat bijvoorbeeld:

```
DEVICE = ANSI.SYS
```

Wijzig die regel in

```
DEVICE = C:\DOS\ANSI.SYS
```

zodat het besturingssysteem de driver weer kan vinden. Wijzig het pad voor de andere verplaatste drivers op dezelfde manier.

COMMAND.COM verplaatsen

U kunt COMMAND.COM ook onderbrengen in de directory C:\DOS. Daartoe wijzigt u in de eerste plaats de verwijzing in het bestand CONFIG.SYS:

```
SHELL = C:\DOS\COMMAND.COM /P
```

In het bestand AUTOEXEC.BAT voegt u de volgende regel toe:

```
SET COMSPEC = C:\DOS\COMMAND.COM
```

Gebruikt u voor uw DOS-bestanden een andere directory, dan vervangt u \DOS\

door de naam van die directory. Vervolgens kopieert u COMMAND.COM naar die directory en wist u het bestand in de hoofddirectory.

MS-DOS heeft er geen problemen mee dat u COMMAND.COM opslaat in een subdirectory, omdat bij het starten van de PC allereerst het bestand CONFIG.SYS wordt gelezen. Daar vindt MS-DOS al de nodige verwijzingen in de SHELL-opdracht. Door de opdracht SET COMSPEC in het bestand AUTOEXEC.BAT weet MS-DOS ook later waar het besturingssysteem te vinden is.

Tip: De ervaring leert dat MS-DOS er wel moeite mee heeft, als u COMMAND.COM zowel in de hoofddirectory opslaat als in een subdirectory, terwijl COMSPEC verwijst naar de subdirectory. Verwijder dus het bestand COMMAND.COM uit de hoofddirectory als u deze tip toepast.

13.5 Het zoekpad uitbreiden

Helaas is het niet zonder meer mogelijk een zoekpad uit te breiden met een nieuwe directory. Steeds als u de opdracht PATH vergezeld van een directory-naam invoert, vervangt die opgave het al bestaande zoekpad. Wilt u het zoekpad uitbreiden, dan dient u het bestaande zoekpad helemaal opnieuw te typen en aan het eind de nieuwe directory toe te voegen.

Het batchbestand ADDPATH.BAT maakt het u wat eenvoudiger:

```
@REM *****************************************************
@REM * batchbestand voor het uitbreiden van het zoekpad *
@REM *****************************************************
@ECHO OFF
PATH %PATH%;%1
ECHO nieuw pad: %PATH%
@ECHO ON
```

Roep het batchbestand op met de naam van de nieuwe directory als parameter. Het batchbestand plaatst dan het nieuwe zoekpad in zijn geheel op uw scherm. Bestaat uw zoekpad tot nu toe bijvoorbeeld alleen uit C:\DOS, en wilt u C:\BAT toevoegen, dan typt u de volgende opdracht:

```
ADDPATH C:\BAT
```

Het batchbestand voert dan de volgende regel uit naar uw scherm:

```
nieuw pad: C:\DOS;C:\BAT
```

Tip: Na een reset is dit zoekpad natuurlijk verloren gegaan. Wilt u het zoekpad blijvend uitbreiden, dan bent u aangewezen op een padopgave in het bestand AUTOEXEC.BAT.

13.6 Gegevens beveiligen tegen per ongeluk wissen

Bepaalde bestanden mogen niet zo maar worden gewist. Denk bijvoorbeeld aan de bestanden in de directory C:\DOS. Gebruik de opdracht ATTRIB om het attribuut *Read only* van dergelijke bestanden te activeren. Daarna kunt u dergelijke bestanden inderdaad alleen nog lezen, totdat u het attribuut weer op nul zet. Probeert u toch zo'n bestand te wissen, dan meldt MS-DOS:

```
Access denied
```

Wilt u het bestand COMMAND.COM beveiligen, dan typt u de volgende opdracht:

```
ATTRIB COMMAND.COM +r
```

Wilt u de beveiliging weer opheffen, dan typt u:

```
ATTRIB COMMAND.COM -r
```

Gebruik de opdracht ATTRIB zonder tweede parameter om een overzicht te krijgen van de bestanden die beveiligd zijn. Bij de uitvoer plaatst MS-DOS een 'R' voor een bestandsnaam als het attribuut Read only is gezet.

U mag bij ATTRIB wildcards gebruiken, zodat u met de hieronder staande opdrachten alle bestanden in de directory C:\DOS in één keer beveiligt.

```
CD C:\DOS
ATTRIB *.* +r
```

Let op! Het bestand DOSSHELL.INI mag u niet op deze manier beveiligen. U zou dan niet goed meer met de DOS Shell kunnen werken, omdat MS-DOS de wijzigingen die u aanbrengt in instellingen, niet kan opslaan.

Een computer heeft een onfeilbaar geheugen en kan dus zonder problemen de juiste syntaxis voor de ATTRIB-opdrachten onthouden, als u de volgende twee batchbestanden, PROTECT.BAT en UNPROTEC.BAT gebruikt.

```
@REM ***************************************************
@REM * batchbestand voor het beveiligen van bestanden *
@REM ***************************************************
@ECHO OFF
IF "%1" == "" GOTO fout
ATTRIB %1 +R
GOTO einde
:fout
ECHO Syntaxis: PROTECT BESTAND.EXT (wildcards toegestaan)
:einde
ECHO ON
```

```
@REM ***********************************************************
@REM *  batchbestand voor het opheffen van de beveiliging van     *
@REM *  bestanden                                                  *
@REM  **********************************************************
@ECHO OFF
IF "%1" == "" GOTO fout
ATTRIB %1 -R
GOTO einde
:fout
ECHO Syntaxis: UNPROTEC BESTAND.EXT (wildcards toegestaan)
:einde
ECHO ON
```

Tip: Steeds meer applicaties gaan ertoe over de configuratie van schermen, toetsen en dergelijke op te slaan in een zogenaamd .INI-bestand. Dergelijke bestanden dient u niet te beveiligen, omdat u anders geen instellingen kunt wijzigen, terwijl u met de applicatie werkt. Wilt u zulke bestanden automatisch uitsluiten in de bovenstaande batchbestanden, dan voegt u aan PROTECT.BAT de volgende regel toe:

```
...
ATTRIB %1 +R
ATTRIB *.INI -R
GOTO einde
...
```

13.7 Abort, Retry, Fail?

Veel foutmeldingen van MS-DOS worden gevolgd door de vraag uit de kop van deze paragraaf. Achter die drie mogelijkheden gaat het volgende schuil.

Abort *Afbreken.* Het programma wordt niet verder afgewerkt, maar onmiddellijk afgebroken. We raden u aan van deze mogelijkheid gebruik te maken als er zich een onduidelijke fout voordoet, die u niet zo maar kunt oplossen. Wilt u bijvoorbeeld een belangrijke tekst kopiëren naar een reservediskette en treedt er een fout op bij het lezen en schrijven van de gegevens, dan kunt u maar het beste afbreken en een andere diskette gebruiken.

Retry *Opnieuw proberen.* Als u de opdracht geeft om bestanden naar een diskette te kopiëren, maar u vergeet een diskette in de drive te doen, dan kunt u de fout 'Not ready reading drive A' opheffen door alsnog een diskette in de drive te plaatsen. Nadat u een 'r' typt wordt de opdracht alsnog uitgevoerd. U kunt de optie ook gebruiken als u belangrijke gegevens op een diskette hebt staan die u wilt lezen, maar MS-DOS meldt een leesfout. Probeer een aantal malen Retry voordat u de moed opgeeft: soms lukt het een tweede of derde keer wel.

Fail *Overslaan.* De opdracht die tot de foutmelding leidt, wordt overgeslagen en DOS gaat verder met de volgende opdracht. De optie is bruikbaar als de niet goed werkende opdracht van ondergeschikt belang is. Geeft u bijvoorbeeld met de opdracht COPY *.* aan DOS op een groot aantal bestanden te kopiëren en bij het zesde bestand verschijnt een foutmelding op uw scherm, dan typt u eerst een paar keer 'R' om het opnieuw te proberen. Helpt dat niet, dan kiest u Fail om het onleesbare bestand over te slaan en de overige bestanden te kopiëren.

Bij leeshandelingen kan DOS u nog een vierde mogelijkheid aanbieden:

Ignore *Negeren.* Als Retry niets oplevert, maar u werkt met onvervangbare gegevens, dan kiest u Ignore. Weliswaar kan DOS dan bepaalde gegevens niet foutvrij lezen, maar de opdracht wordt verder uitgevoerd: de rest van het bestand wordt gelezen. Als het om grote tekstbestanden gaat, is het eenvoudiger een klein beschadigd deel te herstellen, dan het hele bestand opnieuw te maken.

Na Ignore dient u altijd te controleren wat er van de gegevens geworden is, omdat ergens een klein deel beschadigd kan zijn.

13.8 '.' en '..'

Boven in subdirectories vindt u altijd twee regels met punten:

```
.              <DIR>      27-09-91   20:46
..             <DIR>      27-09-91   20:46
```

De enkele punt op de eerste regel is een plaatshouder voor de actuele directory. U kunt dus in plaats van de opdracht DIR ook de opdracht DIR . typen; beide hebben hetzelfde resultaat.

De tweede regel is interessanter. Beide punten staan voor de directory waar de actuele directory een subdirectory van is. Met de opdracht CD.. activeert u dus die directory. Ook in andere opdrachten mag u de beide plaatshouders gebruiken. De opdracht

```
COPY . A:
```

kopieert alle bestanden uit de actuele directory naar de diskette in drive A:. Wilt u alle bestanden in de actuele directory wissen, dan typt u:

```
DEL.
```

Na de controlevraag worden de bestanden gewist. Met de opdracht

```
DIR..
```

haalt u een overzicht van de directory waar de actuele directory een subdirectory van is, naar het scherm.

13.9 Schermuitvoer per pagina

Hoe sneller een PC, hoe meer gemak. Toch gaat dat niet altijd op: een AT met een kloksnelheid van 12 Mhz laat de bestanden van een directory na de opdracht DIR al zo snel over uw scherm vliegen, dat het niet meer te volgen is, laat staan dat u op het juiste moment de waterval tot stoppen brengt.

Bij de opdracht DIR kunt u de stroom gegevens kanaliseren met de optie /P; bij de opdracht TYPE met de optie | MORE. U dient in beide gevallen de juiste optie te onthouden en zonder fouten te typen.

Tijd dus voor het maken van een batchbestand dat het werk voor u doet. Van de drie opdrachten waarbij u |MORE het vaakst zult gebruiken, geven we u hieronder de listing. Om de opdracht van de oorspronkelijke opdracht te kunnen onderscheiden, voegen we aan de naam van het batchbestand een m toe: DIR wordt DIRM.BAT, TYPE wordt TYPEM.BAT en TREE wordt TREEM.BAT.

```
@REM ************************************
@REM *  batchbestand voor: DIR | MORE   *
@REM ************************************
@ DIR %1 %2 %3 | MORE

@REM ************************************
@REM *  batchbestand voor: TYPE | MORE  *
@REM ************************************
@ TYPE %1 | MORE

@REM ************************************
@REM *  batchbestand voor: TREE | MORE  *
@REM ************************************
@ TREE %1 %2 %3 | MORE
```

Na de opdracht TREE zet DOS een overzicht van de directory-structuur op uw scherm.

13.10 Programma's in QBASIC oproepen vanuit AUTOEXEC.BAT

Een groot aantal van de *utilities* in dit boek zijn ondergebracht in batchbestanden, maar een aantal is ondergebracht in een programma in BASIC. Roept u QBASIC samen met het programma op vanuit een batchbestand en sluit u het programma af met SYSTEM in plaats van END, dan wordt de besturing na beëindiging weer overgedragen aan het batchbestand, dat dan verder wordt afgewerkt.

Op die manier kunt u ook standaardhandelingen bij het starten inbouwen in het bestand AUTOEXEC.BAT. We zullen dat demonstreren aan de hand van een voorbeeldprogramma, dat u helpt herinneren op tijd een nieuwe backup te maken van de gegevensbestanden op de harde schijf. Maak met behulp van de tekstverwerker van QBASIC het volgende programmabestand BACKDAT.BAS.

```
REM Geheugensteun voor het maken van een backup
```

```
REM formaat-datum: mm-dd-jjjj
OPEN "c:\bat\backup.dat" FOR INPUT AS #1
LINE INPUT #1, back$
IF back$ = DATE$ THEN
  BEEP
  PRINT "U had u voorgenomen vandaag een backup te maken."
  PRINT "Vandaag is het "; back$
  PRINT "Druk op een toets..."
  WHILE INKEY$ = "" : WEND
END IF
CLOSE #1
SYSTEM
```

Met behulp van de opdracht COPY CON maakt u het bestand BACKUP.DAT aan, waarin u in het juiste formaat de datum invoert waarop u een backup wilt maken. Om het juiste formaat te achterhalen voor QBASIC, typt u (terwijl QBASIC actief is) de opdracht:

```
? DATE$
```

Vervolgens voegt u aan het bestand AUTOEXEC.BAT de volgende regel toe:

```
QBASIC /RUN BACKDAT.BAS
```

Op de dag die u hebt opgegeven, wordt u bij het starten gewaarschuwd met een pieptoon en de bijbehorende boodschap.

Tip: Dit programma is niet bepaald een fraai voorbeeld van hoe het moet: Zou u op de opgegeven dag uw computer niet aanzetten, dan werkt het uiteraard volstrekt niet. U dient dan een routine in te bouwen die voor u uitzoekt of de bewuste dag nog moet komen of misschien al voorbij is. We hebben dit programma gebruikt om te laten zien hoe u programma's in QBASIC koppelt aan het bestand AUTOEXEC.BAT.

13.11 Diskettes kopiëren via de harde schijf

Als u een PC hebt met een harde schijf en één diskettestation, is het kopiëren van diskettes een omvangrijk karwei, zeker als het diskettestation een capaciteit heeft van bijvoorbeeld 1,44 Mb. Het werkgeheugen van de PC is dan namelijk te klein om alle gegevens in één keer op te slaan, zodat u steeds bron- en doeldiskettes moet verwisselen.

Het hieronder staande batchbestand DCOPY.BAT helpt u daarbij als u aan de volgende voorwaarden voldoet:

- De doeldiskette moet zijn geformatteerd.
- De brondiskette mag niet zijn onderverdeeld in subdirectories, want die worden niet meegekopieerd.
- U dient in de directory C:\BAT te beschikken over een bestand YES. U kunt dat makkelijk aanmaken op de volgende manier:

```
COPY CON C:\BAT\YES
Y<Enter>
<Ctrl-Z><Enter>
```

De opdrachten <Ctrl-Z> en <Enter> voert u letterlijk uit door op de desbetreffende toetsen te drukken.

- De doeldiskette dient leeg te zijn. Als dat niet het geval is, is er niet genoeg ruimte op de diskette, omdat de bestanden niet over de al bestaande bestanden heen worden geschreven.

Het batchbestand ziet er als volgt uit:

```
@REM ***************************************************************
@REM *  Batchbestand voor het kopiëren met behulp van de harde      *
@REM *  schijf                                                       *
@REM ***************************************************************
@ECHO OFF
@IF NOT EXIST C:\DCOPY MD C:\DCOPY
@IF EXIST C:\DCOPY DEL <C:\BAT\YES C:\DCOPY\*.*
@ECHO Plaats de brondiskette in de drive en druk op een toets...
@PAUSE
@COPY A:\*.* C:\DCOPY
@ECHO Plaats de doeldiskette in de drive en druk op een toets...
@PAUSE
@COPY C:\DCOPY\*.* A:\
@DEL C:\BAT\YES C:\DCOPY\*.*
@RD C:\DCOPY
@ECHO Klaar
@ECHO ON
```

Start het programma met DCOPY en plaats de brondiskette in drive A:. Nadat u op een willekeurige toets drukt, begint het programma met kopiëren van de bestanden naar een speciaal daarvoor aangemaakte directory C:\DCOPY. Daarna vraagt het

programma u de brondiskette in de drive te plaatsen, zodat het alle bestanden vanuit C:\DCOPY naar de doeldiskette kan kopiëren. Ten slotte wordt de directory C:\DCOPY schoongemaakt en gewist.

Tip: Verander het bestand een klein beetje om de bestanden van de doeldiskette te wissen:

```
@ECHO Plaats de doeldiskette in de drive en druk op een toets...
@ECHO De bestanden van de doeldiskette worden gewist!
@PAUSE
@DEL C:\BAT\YES A:\*.*
```

13.12 Tips voor het gebruik van DEL *.*

Gebruik wildcards bij het wissen met de nodige terughoudendheid, want ze kunnen ongewild gegevens wissen. Met name twee situaties die zich kunnen voordoen, leiden nogal eens tot rampen:

1 Veronderstel dat u een directory hebt met onder andere de volgende bestanden:

```
BOEK1.TXT
BOEK1.BAK
BOEK2.TXT
BOEK2.BAK
BOEKGES.TXT
BOEKGES.BAK
```

U wilt de bestanden wissen en geeft dus de opdracht DEL BOEK*.*. Achteraf blijkt dat u ook de belangrijke bestanden BOEKIDEE.TXT en BOEKCORR.TXT hebt gewist, waarvan u dacht dat ze in een andere directory stonden. Vergewis u er dus eerst met de opdracht DIR van dat de opdracht om te wissen geen ongewenste neveneffecten heeft. Als u al wat vertrouwder bent met het maken van batchbestanden, maakt u daarvoor natuurlijk een mooi batchbestand aan onder de naam DEL.BAT... (zie ook hoofdstuk acht).

2 Een veel voorkomende situatie is deze: u wilt in een directory alle bestanden wissen en gebruikt de opdracht CD om de directory te activeren. Vervolgens geeft u de opdracht DEL.. U verwacht de controlevraag en routinematig drukt u op 'y'. Achteraf komt u erachter dat u alle bestanden van de hoofddirectory hebt gewist...

Het lijkt onbegrijpelijk, maar is dat in het geheel niet. Laten we ervan uitgaan dat het om de directory C:\TEMP gaat, die u wilt opschonen. In de haast hebt u echter getypt:

```
CD C:\TEMO
```

De 'o' zit vlak naast de 'p' op uw toetsenbord. Het resultaat is een foutmelding van DOS: 'Invalid directory'. Er staat zo veel op uw scherm, het valt u niet eens op. Het gevolg is echter dat de hoofddirectory actief blijft en dat de volgende opdracht DEL. daar wordt uitgevoerd. Als u geluk hebt, ontdekt u de fout meteen en kunt u met UNDELETE de schade tot een minimum beperken.

Het onderstaande geval is vergelijkbaar. De actuele directory is C:\DBDATA. U bewaart er al uw dBASE-bestanden. U bent bovendien behept met de goede gewoonte iedere dag van uw gegevensbestanden een reservekopie te maken op een diskette in drive A:. Per ongeluk typt u echter de volgende opdrachten:

```
A
DEL *.*
Y
```

U bedoelt de achterhaalde bestanden op de diskette te wissen; het resultaat is dat u de actuele bestanden wist en alleen nog beschikt over de achterhaalde. Ook in dit geval hebt u routinematig de foutmelding over het hoofd gezien. Als u daarna de opdracht

```
COPY *.* A:\
```

geeft, zal het systeem tot uw verbazing melden: '0 file(s) copied'. U moet al heel veel slaap hebben op dat moment, als die melding u ontgaat. Normaal gesproken immers verschijnt er een lijstje met bestanden op uw scherm die worden gekopieerd. Op dat moment kunt u de schade ook nog weer zo veel mogelijk herstellen met UNDELETE. U kunt de situatie nog beter vermijden met een macro die de foutieve opdracht verandert in een correcte opdracht: zie paragraaf 5.3.

13.13 AUTOEXEC.BAT en CONFIG.SYS beveiligen

Goede software wordt tegenwoordig geleverd met een installatieprogramma. Dat maakt directories voor u aan, kopieert er de bestanden naar toe en past de configuratie van uw systeem zodanig aan dat het programma optimaal kan presteren.

U hoeft daar niet altijd even gelukkig mee te zijn. Als een applicatie er geen weet van heeft dat u, omdat u een cache-geheugen gebruikt, het aantal buffers hebt teruggebracht naar drie, kan het zijn dat de installatieprocedure weer 20 buffers installeert. Het cache-geheugen werkt dan juist niet meer optimaal.

U dient dus eerst de bestanden AUTOEXEC.BAT en CONFIG.SYS te beveiligen. Dat kan als u voor beide het attribuut *Read only* zet. Daarna kunnen de bestanden alleen nog worden gelezen, niet meer gewijzigd. Als de bestanden op de harde schijf staan, gaat dat met de volgende opdrachten:

```
ATTRIB C:\AUTOEXEC.BAT +R
ATTRIB CONFIG.SYS +R
```

Een installatieprocedure kan nu niet meer ongemerkt uw bestanden wijzigen. Hij zal een foutmelding uitvoeren naar het scherm, zodat u maatregelen kunt treffen: kopieer de bestanden naar een veilige plaats en hef pas dan de beveiliging op met de opdrachten:

```
ATTRIB C:\AUTOEXEC.BAT -R
ATTRIB CONFIG.SYS -R
```

Vergelijk na de installatie de bestanden en besluit vervolgens pas of de wijzigingen u bevallen.
Een bijzonder gevaarlijke wijziging is de opdracht:

```
SET COMSPEC = C:\COMMAND.COM
```

Misschien werkt u met meer dan een besturingssysteem (MS-DOS in verschillende versies) en bevindt de juiste versie zich in een subdirectory. De wijziging die het gevolg is van de opdracht SET COMSPEC kan dan al gauw catastrofale gevolgen hebben: het systeem werkt niet goed meer of blijft zelfs helemaal hangen. De foutmeldingen

```
Invalid COMMAND.COM
```

en

```
Cannot load COMMAND.COM. System halted
```

leiden er niet zelden toe dat u een paar uur extra tijd moet steken in het herstellen van een beschadigde configuratie.

13.14 Regels tellen in een bestand

Als u graag wilt weten hoeveel regels in uw bestand (bijvoorbeeld een tekstbestand) staan, kunt u de volgende opdracht typen:

```
FIND /c " " BESTAND.EXT
```

De opdracht zoekt in het bestand naar spaties en voert uiteindelijk door toedoen van de parameter /c het aantal regels uit waarin spaties zijn gevonden.

Als een regel echter geen spaties bevat, is het resultaat niet correct: witregels in tekstbestanden die worden gemarkeerd door een stuurteken, bevatten geen spaties. Het hieronder staande batchbestand COUNT.BAT verdient daarom de voorkeur.

```
@REM ************************************************************
@REM * Batchbestand voor het tellen van regels in een bestand *
@REM ************************************************************
@ECHO OFF
IF "%1" == "" GOTO fout
FIND /c /v "!Dummytext!" %1
GOTO einde
:fout
ECHO Syntaxis: COUNT BESTAND.EXT
:einde
@ECHO ON
```

Het bestand telt alle regels waarin de zoekstring '!Dummytext!' niet voorkomt. U mag natuurlijk ook als zoekstring 'Mijn grootmoeder heet Roodkapje' gebruiken: het resultaat zal hetzelfde zijn, want beide zoekstrings zijn behoorlijk ongebruikelijk.

13.15 Opdrachten uitvoeren voor alle bestanden van een directory

Er zijn nogal wat DOS-opdrachten waarbij u geen wildcards mag gebruiken. Een daarvan is bijvoorbeeld de opdracht TYPE, maar ook de opdracht FIND die we in de vorige paragraaf hebben gebruikt, hoort daarbij. Met behulp van de DOS-opdracht FOR...DO kunt u een batchbestand maken dat het probleem uit de wereld helpt.

Wilt u bijvoorbeeld alle regels tellen in de batchbestanden van de directory C:\BAT. dan typt u de volgende opdracht:

```
FOR %a IN (C:\BAT\*.BAT) DO FIND /c /v "Sneeeeuwwitje" %a
```

Alle bestandsnamen worden nu met opgave van het aantal regels opgesomd. Probeert u het eens uit. Niet alleen is het een heidens karwei om de opdracht zonder fouten te typen, u zult hem voor iedere directory opnieuw moeten typen (of wijzigen als u DOSKEY gebruikt). Met behulp van het batchbestand DO.BAT gaat het makkelijker en mooier:

```
@REM **************************************************
@REM * Batchbestand voor het uitvoeren van opdrachten *
@REM *     voor alle bestanden van een directory      *
@REM **************************************************
@ECHO OFF
IF "%1" == "" GOTO fout
IF "%2" == "" GOTO fout
@FOR %%a IN (%2) DO %1 %%a %3 %4 %5
GOTO einde
:fout
ECHO Syntaxis: DO BESTAND.EXT PARAMETERS
:einde
@ECHO ON
```

Als u nu de opdracht

```
DO TYPE *.BAT
```

typt, is het resultaat een overzichtelijke weergave van de inhoud van alle bestanden van de actuele directory. De volgende opdracht wist alle bestanden in de actuele directory zonder controlevraag!

```
DO DEL *.*
```

13.16 Alle bestanden snel naar het scherm of de printer

De onderstaande opdracht zou u ook met het batchbestand uit de vorige paragraaf kunnen uitvoeren, al dient u dan wel voor iedere opdracht een apart batchbestand te schrijven. Met de opdracht COPY kunt u echter eenvoudig alle bestanden van een directory naar een randapparaat dirigeren. Wilt u bijvoorbeeld alle batchbestanden van de actuele directory uitvoeren naar het scherm, dan typt u:

```
COPY *.BAT CON
```

CON staat namelijk bij invoer voor het toetsenbord en bij uitvoer voor het scherm. U kunt de bestanden even gemakkelijk naar de printer sturen met de opdracht:

```
COPY *.BAT LPT1
```

13.17 Bevestigingen van opdrachten overslaan

Er zijn veel DOS-opdrachten die om een bevestiging vragen of anderszins een reactie van de gebruiker verwachten. Op zijn minst dient u dan op Enter te drukken, zoals bijvoorbeeld bij de opdrachten DATE en TIME. Als u die opdrachten in een batchbestand wilt gebruiken dat automatisch een serie opdrachten uitvoert, zult u een manier moeten vinden dergelijke bevestigingen te omzeilen. Dat kan door ze na te bootsen in een klein bestand dat u met het pipe-symbool voor invoer '<' activeert in het batchbestand. Vooral 'Yes', 'No' en 'Enter' lenen zich daarvoor. Maak in de directory C:\BAT drie kleine tekstbestanden aan volgens de onderstaande listings. Voer de opdrachten tussen punthaken steeds letterlijk uit.

```
COPY CON YES<Enter>
Y<Enter>
<Ctrl-Z><Enter>

COPY CON NO<Enter>
N<Enter>
<Ctrl-Z><Enter>

COPY CON ENTER
<Enter>
<Ctrl-Z><Enter>
```

Als u vervolgens het actuele tijdstip wilt opnemen in een bestand, kunt u de bestanden op de volgende manier gebruiken:

```
TIME >TIME.DAT <C:\BAT\ENTER
```

De uitvoer van de opdracht TIME die normaal gesproken naar het scherm wordt gestuurd, wordt omgeleid naar het bestand TIME.DAT. De invoer, namelijk de toetsaanslag Enter, haalt de opdracht uit het bestand C:\BAT\ENTER.

Hebt u een batchbestand dat alle bestanden van een directory moet wissen zonder

dat u eerst de opdracht bevestigt, dan gebruikt u de opdracht DEL *.* in combinatie met het invoerbestand C:\BAT\YES.

Wilt u niet eens de meldingen van de wisprocedure op uw scherm zien, dan kunt de uitvoer nog naar het imaginaire randapparaat NUL sturen. De opdracht ziet er dan als volgt uit:

```
DEL <C:\BAT\YES >NUL C:\TEMP\*.*
```

Een derde gebruiksmogelijkheid versnelt het werken met applicaties die veel gebruik maken van zogenaamde *overlay-bestanden.* Als zo'n applicatie iedere keer een bestand van schijf moet lezen, moet u steeds wachten. Maakt u een batchbestand dat bij het starten alle bestanden naar een RAM-disk kopieert, en start u het programma vanaf die RAM-disk, dan kan het programma met elektronische snelheid zijn gegevens ophalen.

Wilt u echter op dezelfde manier een tweede applicatie starten, dan blijkt dat uw RAM-disk vol is. Om dat te voorkomen, neemt u in het batchbestand van alle programma's die gebruik maken van de RAM-disk de volgende opdracht op:

```
DEL <C:\BAT\YES >NUL E:\*.*
```

Voordat nu de nieuwe bestanden worden geladen, wordt de RAM-disk eerst helemaal schoongemaakt. Voor het computerspel SIMCITY ziet zo'n batchbestand er bijvoorbeeld als volgt uit:

```
@ECHO OFF
DEL. <C:\BAT\YES >NUL E:\*.*
E:
XCOPY A:\*.*
SIMCITY
@ECHO ON
```

Let op: Als het programma zelf gegevensbestanden aanmaakt of wijzigt, dient u ervoor te zorgen dat die niet naar de RAM-disk worden geschreven, want dan gaan ze later verloren.

13.18 Een koude start vanuit een batchbestand

Een koude start houdt in dat uw systeem helemaal vanaf de grond af opnieuw wordt geconfigureerd. Normaal gesproken gebruikt u daarvoor de aan/uit-schakelaar. Het

kan ook met behulp van een batchbestand dat een klein programma oproept in machinetaal.

Het programma in machinetaal BOOT.DEB maakt u aan met behulp van een tekstverwerker, bijvoorbeeld EDIT. U voert de volgende regels letterlijk in en sluit iedere regel af met Enter.

```
A
JMP F000:FFF0

RCX
5
nBOOT.COM
W
Q
```

Nadat u het bestand hebt opgeslagen start u DEBUG met de volgende opdracht:

```
DEBUG <BOOT.DEB
```

Vervolgens beschikt u over een uitvoerbaar programma. Zorg ervoor dat u geen belangrijke bestanden meer in de RAM-disk hebt staan of dat er anderszins gegevens verloren kunnen gaan, dan kunt u het uitproberen door achter de DOS-prompt de volgende opdracht te typen:

```
BOOT
```

Sluit af met Enter. Het resultaat is een reset van het systeem.

13.19 De environment space uitbreiden

Als u veelvuldig gebruik maakt van systeemvariabelen, kan het voorkomen dat u bijvoorbeeld na een opdracht met SET geconfronteerd wordt met de foutmelding:

```
Out of environment space
```

U kunt het besturingssysteem MS-DOS namelijk aan uw wensen aanpassen met behulp van systeemvariabelen die worden opgeslagen in een gereserveerd gedeelte van het geheugen, de environment space. De belangrijkste systeemvariabelen zijn:

PATH alle directories waarin DOS zoekt naar een bestand

PROMPT de weergave van de systeemprompt

COMSPEC zoekpad naar en naam van het besturingssysteem

Als u nu meer systeemvariabelen definieert (met de opdracht SET) of u breidt bijvoorbeeld het zoekpad uit, kan het zijn dat de gereserveerde geheugenruimte niet meer toereikend is.

U breidt die geheugenruimte uit door het besturingssysteem opnieuw op te roepen met de parameter /E:xxxx. 'xxxx' staat daarbij voor het aantal bytes dat moet worden gereserveerd. Bij verstek reserveert MS-DOS 160 bytes.

Als het probleem zich vaak voordoet, heeft het zin te zorgen voor een blijvende oplossing. Dat doet u met behulp van de opdracht SHELL in het bestand CONFIG.SYS. Als uw COMMAND.COM in de directory C:\DOS staat en u wilt de environment space blijvend instellen op 1000 bytes, voegt u aan het bestand CONFIG.SYS de volgende regel toe:

```
SHELL = C:\DOS\COMMAND.COM /P /E:1000
```

De parameter /P geeft aan dat het besturingssysteem permanent moet worden geïnstalleerd en het oude besturingssysteem moet vervangen.

13.20 Schermuitvoer via PROMPT

Sommige gebruikers houden van een sober scherm met weinig tierelantijnen. Anderen laten DOS het liefst alle mogelijke soorten van gegevens op het scherm zetten. Als u tot de laatste groep behoort, kunt u uw hart ophalen met de mogelijkheden van de opdracht PROMPT. De volgende opdracht plaatst tijdstip, datum en actueel zoekpad op het scherm:

```
PROMPT $t $d $p$g
```

$t zorgt voor de weergave van de tijd, $d voor de weergave van de datum en $p voor de weergave van de drive en het zoekpad. Het groter-dan teken dat de prompt afsluit, wordt gegenereerd met $g. Probeert u het eens uit. Het resultaat is een lange, onoverzichtelijke systeemprompt.

We hebben een batchbestand voor u bedacht, dat dezelfde gegevens naar de bovenste regel van het scherm uitvoert en daar invers afdrukt. We noemen het batchbestand KOPREGEL.BAT:

```
@REM ****************************************************
@REM * Kopregel met tijd, datum en zoekpad (via ANSI.SYS) *
@REM ****************************************************
@PROMPT $e[s$e[f$e[7m$t$h$h$h$h$h$h tijd $d $p$g$e[K$e[u$e[0m$p$g
```

Wilt u deze weergave van de systeemprompt weer verwijderen, dan voert u de opdracht PROMPT nogmaals in, nu met de parameters die uw eigen systeemprompt oproepen, bijvoorbeeld:

```
PROMPT $p$g
```

Het batchbestand werkt alleen als u in het bestand CONFIG.SYS de device driver ANSI.SYS hebt geactiveerd. Dat doet u met de volgende opdrachtregel:

```
DEVICE = C:\DOS\ANSI.SYS
```

Als u het programma in een ander directory hebt opgeslagen, vervangt u het zoekpad C:\DOS\ door het zoekpad naar die directory. Bij de opdracht PROMPT mag u een hele serie parameters gebruiken, die we hieronder kort zullen toelichten.

$e[De aankondiging van een opdracht voor de device driver ANSI.SYS. $e wordt door het besturingssysteem vertaald in 'Escape'.

$e[s Slaat de actuele positie van de cursor op.

$e[f Plaatst de cursor op de aangegeven positie: $e[f10;20 plaatst de cursor bijvoorbeeld op regel tien in kolom twintig. Geeft u geen regel en kolom op, dan wordt de cursor in de linker bovenhoek van het scherm geplaatst.

$e[7m Inversie van alle eropvolgende uitvoer naar het scherm.

$t Uitvoer naar het scherm van de actuele tijd.

$h Bootst de toetsaanslag Backspace na en zorgt er dus voor dat het vorige teken wordt gewist en dat de cursor een kolom naar links wordt verplaatst. In het bovenstaande voorbeeld worden bij de uitvoer van de tijd de seconden en honderdsten van seconden onderdrukt.

$d Uitvoer naar het scherm van de datum.

pg Uitvoer naar het scherm van de actuele drive en het zoekpad, afgesloten met het groter-dan teken.

$e[K Wist de rest van de regel.

$e[u Plaatst de cursor weer op de positie die eerder is opgeslagen met de reeks **$e[s**.

$e[0m Heft de inversie van de uitvoer naar het scherm op.

13.21 Gegevens van de systeemconfiguratie bewaren

U hebt een AT met een harde schijf die prima functioneert. Om ervoor te zorgen dat dat ook zo blijft, zou u eens naar de systeemconfiguratie moeten kijken. In iedere AT is een SETUP-programma ingebouwd met essentiële gegevens over uw computer. Die gegevens blijven bewaard in het geheugen, ook nadat u de computer uitschakelt. Een batterij zorgt voor de nodige voeding.

Raakt de batterij onverhoopt leeg, dan start u de computer, maar zijn de gegevens over de harde schijf, het soort scherm en toetsenbord dat u gebruikt, niet meer voorhanden. Er verschijnt een foutmelding op uw scherm samen met een toetsencombinatie die het SETUP-programma activeert.

De toetsencombinatie waarmee u het SETUP-programma oproept, verschilt van systeem tot systeem. In veel gevallen flitst tijdens het starten een aanwijzing over uw scherm, bijvoorbeeld 'Press (DEL) to run SETUP'. Andere systemen activeren SETUP met de combinatie Ctrl-Alt-Esc of Ctrl-Alt-S.

Het loont de moeite de configuratiegegevens over te nemen. Er zijn meer dan veertig typen harde schijven en de documentatie bij die harde schijven is niet altijd even makkelijk te achterhalen. De gegevens

```
                                Cyl  Hd  Pre  LZ   Sec  Size
Hard Disk 1:     Type 32        615   8  128  664   17   40
```

onthoudt u niet uit het blote hoofd. Toch hebt u ze nodig om de SETUP weer compleet te maken als u beschikt over dit type harde schijf.

Tip: Zie de handleiding bij uw computer voor de juiste toetsencombinatie om SETUP te starten.

13.22 Een screensaver in QBASIC

Laat u de computer een tijdje aan staan, zonder dat u er gebruik van maakt, dan verandert de weergave op het scherm niet. Op den duur bestaat het gevaar dat daardoor de tekens inbranden en dus de kwaliteit van de schermweergave achteruit gaat. Er zijn veel mogelijkheden om dit probleem op te lossen; een heel eenvoudige methode is het hieronder staande programmaatje in QBASIC. Voer het programma in onder de naam SCREEN.BAS:

```
REM Weergave op het scherm uitschakelen
CLS
WHILE INKEY$ = "" : WEND
SYSTEM
```

Vanuit DOS roept u het programma op met de opdracht:

```
QBASIC /RUN SCREEN.BAS
```

Wilt u daarna weer aan het werk, dan drukt u eenvoudigweg op een toets. Het programma wordt beeindigd en de besturing wordt met de opdracht SYSTEM weer overgedragen aan MS-DOS. Vanzelfsprekend werkt het programma nog beter als u de opdracht waarmee u het programma oproept, opslaat in een batchbestand SCREEN.BAT.

Tip: Beschikt u over een langzame PC, dan stoort het dat eerst QBASIC moet worden geladen. In dat geval kunt u ook een klein batchbestand aanmaken met hetzelfde doel. De enige voorwaarde is dat de driver ANSI.SYS is geïnstalleerd. Het batchbestand SCREEN.BAT ziet er als volgt uit:

```
@SET Oldprompt = %prompt%
@PROMPT $e[8m
CLS
@PAUSE >NUL
@PROMPT $e[0m
SET PROMPT = %oldprompt%
@SET OLDPROMPT=
CLS
```

13.23 Een leeg diskettestation verlaten

Geeft u de opdracht drive A: te activeren zonder er een diskette in te plaatsen, dan verschijnt de foutmelding:

```
Not ready reading drive A
Abort, Retry, Fail?
```

U kunt nu net zo vaak als u maar wilt, een 'A' of een 'R' typen, de vraag 'Abort, Retry, Fail?' zal steeds opnieuw op uw scherm verschijnen. U dient de derde optie 'Fail' te kiezen. Er verschijnt een nieuwe melding op uw scherm:

```
Current drive is no longer valid
```

U dient de oude drive te herstellen door achter de systeemprompt bijvoorbeeld C: te typen. U kunt ook alsnog een diskette in A: plaatsen en vervolgens opnieuw A: achter de systeemprompt typen.

13.24 De opdrachtregel wissen

Een beetje gecompliceerde opdracht bestaat in DOS al gauw uit zo'n 30 tot 40 tekens. U typt hem helemaal en bedenkt voor u op Enter drukt dat het toch eigenlijk maar anders moet. Met de Backspace-toets kunt u nu alles wissen, maar het kan ook sneller.

Drukt u op Esc, dan wordt de regel ook in zijn geheel gewist en kunt u opnieuw een opdracht invoeren. Als u met DOSKEY werkt en u hebt de opdracht die u met Esc wilt wissen, opgehaald met cursor omhoog, dan blijft die opdracht wel bewaard in de buffer van DOSKEY.

Register

DOS 5 Tips & Trucs

Grensverleggende Gebruikersgeheimen

De meeste gebruikers kennen het verschijnsel dat onder het oppervlak van het dagelijkse gebruik van software vaak nog veel meer mogelijkheden schuilgaan dan men vermoedt. Lang niet al die mogelijkheden en slimme handigheidjes zijn gedocumenteerd door de softwarefabrikant, terwijl kennis van deze *tips & trucs* het werken vaak aanzienlijk vergemakkelijkt.

DOS 5 *Tips & Trucs* leert u de verborgen geheimen van dit nieuwste operating system. Bijvoorbeeld hoe u het beschikbare geheugen onder DOS optimaal benut en hoe u de snelheid van uw PC kunt opvoeren. Duidelijk en vooral gericht op de praktijk, waardoor DOS 5 *Tips & Trucs* toegankelijk is voor een brede groep gebruikers.

Uit de inhoud:
- Tips voor batchbestanden
- DOS in uw eigen huisstijl
- De Task swapper laden zonder wachten
- Optimaal geheugenbeheer
- Opdrachten uitvoeren voor alle bestanden van een schijf
- Snel Backups maken
- Bevestigingen van opdrachten overslaan
- Een screensaver in QBasic.

Kortom, een schat aan handige, nuttige en praktische informatie voor alle DOS 5-gebruikers.

ISBN 90-229-3778-X



Systeemvereisten: een IBM-compatibele PC/XT/AT/386/486 en MS-/PC-DOS 5.0 (wordt niet meegeleverd).